09-CV-00681-CMP

Summons issued SEA 26217

UNITED STATES DISTRICT COURT
WESTERN DISTRICT OF WASHINGTON
AT SEATTLE

AMAZON.COM, INC.,

Plaintiff,

v.

DISCOVERY COMMUNICATIONS, INC.,

Defendant.

No. C09-0681 RSL

COMPLAINT FOR PATENT INFRINGEMENT

JURY DEMAND

Plaintiff Amazon.com, Inc. ("Amazon.com"), by and through its attorneys, alleges for its Complaint against defendant Discovery Communications, Inc. ("Discovery") the following:

NATURE OF THE ACTION

1. This is a civil action for patent infringement brought pursuant to the patent laws of the United States, Title 35 of the United States Code.

PARTIES

2. Plaintiff Amazon.com is a corporation duly organized and existing under the laws of Delaware, and it has a principal place of business at 1200 Twelfth Avenue South,

CORR CRONIN MICHELSON
BAUMGARDNER & PREECE LLP
1001 Fourth Avenue, Suite 3900
Seattle, Washington 98154-1051
Tel (206) 625-8600
Fax (206) 625-0900

ORIGINAL

Seattle, Washington. Amazon.com is a leading online retailer providing consumers with the largest selection of products available for online purchase through the Internet. Since its launch in 1995, Amazon.com has been a pioneer in the field of electronic commerce ("ecommerce"). Amazon.com's creative technological solutions in this field have resulted in the United States Patent and Trademark Office's issuing dozens of patents to Amazon.com for its innovations. These patents relate to, *inter alia*, Amazon.com's novel contributions to the fundamental technology underlying ecommerce.

3. On information and belief, defendant Discovery is a corporation organized under the laws of Delaware and with a principal place of business in Silver Spring, Maryland.

JURISDICTION AND VENUE

4. This is an action alleging patent infringement arising under Title 35 of the United States Code. This court has jurisdiction over the subject matter of this action pursuant to 28 U.S.C. §§ 1331 and 1338(a).

5. This Court has personal jurisdiction over Discovery because on information and belief, Discovery has transacted business in this district, contracted to supply goods or services in this district directly or through its agents, purposely availed itself of the privileges and benefits of the laws of the state of Washington, and committed acts of patent infringement during the course of its business in this district.

6. Venue is proper in this district under 28 U.S.C. §§ 1391(b) and (c) and 1400(b).

CLAIMS FOR RELIEF

7. Amazon.com incorporates by reference Paragraphs 1 through 5 above as if fully set forth herein.

CORR CRONIN MICHELSON
BAUMGARDNER & PREECE LLP
1001 Fourth Avenue, Suite 3900
Seattle, Washington 98154-1051
Tel (206) 625-8600
Fax (206) 625-0900

First Claim for Relief for Infringement of U.S. Patent No. 6,006,225

8. On December 21, 1999, U.S. Patent No. 6,006,225, entitled "Refining Search Queries by the Suggestion of Correlated Terms from Prior Searches," (the "'225 Patent") was duly and legally issued to Amazon.com. Amazon.com owns, by valid assignment, all rights, title, and interest in the '225 patent. A copy of the '225 Patent is attached to this Complaint as Exhibit 1.

9. Discovery owns, operates, provides, receives revenue from, and/or is otherwise affiliated with an online store that is accessible at http://store.discovery.com ("the Discovery Store website") that sells products related to, for example, the Discovery Channel, The Learning Channel, Animal Planet, the Military Channel, and the Science Channel.

10. Discovery has been, currently is, and will continue to directly and/or indirectly infringe, solely or jointly with others, or induce others to infringe one or more claims of the '225 Patent by directly or indirectly, individually or jointly, using or causing to be used Amazon.com's patented system and method through at least the operation of the Discovery Store website. Discovery's infringing activities violate 35 U.S.C. § 271.

11. As a direct and proximate consequence of Discovery's infringement of the '225 Patent, Amazon.com has suffered and will continue to suffer irreparable injury and damages.

Second Claim for Relief for Infringement of U.S. Patent No. 6,169,986

12. On January 2, 2001, U.S. Patent No. 6,169,986 entitled "System and Method for Refining Search Queries," ("the '986 Patent") was duly and legally issued to Amazon.com. Amazon.com owns, by valid assignment, all rights, title, and interest in the '986 Patent. A copy of the '986 Patent is attached to this Complaint as Exhibit 2.

13. Discovery has been, currently is, and will continue to directly and/or indirectly

CORR CRONIN MICHELSON
BAUMGARDNER & PREECE LLP
1001 Fourth Avenue, Suite 3900
Seattle, Washington 98154-1051
Tel (206) 625-8600
Fax (206) 625-0900

infringe, solely or jointly with others, or induce others to infringe one or more claims of the '986 Patent by directly or indirectly, individually or jointly, using or causing to be used Amazon.com's patented system and method through at least the operation of the Discovery Store website. Discovery's infringing activities violate 35 U.S.C. § 271.

14. As a direct and proximate consequence of Discovery's infringement of the '986 Patent, Amazon.com has suffered and will continue to suffer irreparable injury and damages.

Third Claim for Relief for Infringement of U.S. Patent No. 6,266,649

15. On July 24, 2001, U.S. Patent No. 6,266,649 entitled "Collaborative Recommendations Using Item-to-Item Similarity Mappings," ("the '649 Patent") was duly and legally issued to Amazon.com. Amazon.com owns, by valid assignment, all rights, title, and interest in the '649 Patent. A copy of the '649 Patent is attached to this Complaint as Exhibit 3.

16. Discovery has been, currently is, and will continue to directly and/or indirectly infringe, solely or jointly with others, or induce others to infringe one or more claims of the '649 Patent by directly or indirectly, individually or jointly, using or causing to be used Amazon.com's patented system and method through at least the operation of the Discovery Store website. Discovery's infringing activities violate 35 U.S.C. § 271.

17. As a direct and proximate consequence of Discovery's infringement of the '649 Patent, Amazon.com has suffered and will continue to suffer irreparable injury and damages.

Fourth Claim for Relief for Infringement of U.S. Patent No. 6,317,722

18. On November 13, 2001, U.S. Patent No. 6,317,722 entitled "Use of Electronic Shopping Carts to Generate Personal Recommendations," ("the '722 Patent") was duly and

CORR CRONIN MICHELSON
BAUMGARDNER & PREECE LLP
1001 Fourth Avenue, Suite 3900
Seattle, Washington 98154-1051
Tel (206) 625-8600
Fax (206) 625-0900

legally issued to Amazon.com. Amazon.com owns, by valid assignment, all rights, title, and interest in the '722 Patent. A copy of the '722 Patent is attached to this Complaint as Exhibit 4.

19. Discovery has been, currently is, and will continue to directly and/or indirectly infringe, solely or jointly with others, or induce others to infringe one or more claims of the '722 Patent by directly or indirectly, individually or jointly, using or causing to be used Amazon.com's patented system and method through at least the operation of the Discovery Store website. Discovery's infringing activities violate 35 U.S.C. § 271.

20. As a direct and proximate consequence of Discovery's infringement of the '722 Patent, Amazon.com has suffered and will continue to suffer irreparable injury and damages.

PRAYER FOR RELIEF

WHEREFORE, plaintiff Amazon.com prays for judgment and at least the following relief:

A. A judgment declaring that Discovery has infringed one or more claims of the '225 Patent, the '986 Patent, the '649 Patent, and the '722 Patent in violation of 35 U.S.C. § 271, and that those patents are valid and enforceable;

B. For preliminary and permanent injunctive relief restraining and enjoining Discovery and its officers, agents, servants, employees, attorneys, and those persons in active concert or participation with it who receive actual notice of the order by personal service or otherwise, from any further infringement of the '225 Patent, the '986 Patent, the '649 Patent, and the '722 Patent;

C. Damages sufficient to compensate Amazon.com for Discovery's infringement of the '225 Patent, the '986 Patent, the '649 Patent, and the '722 Patent;

CORR CRONIN MICHELSON
BAUMGARDNER & PREECE LLP
1001 Fourth Avenue, Suite 3900
Seattle, Washington 98154-1051
Tel (206) 625-8600
Fax (206) 625-0900

D. An award of pre-judgment and post-judgment interest, and costs, expenses and attorneys' fees pursuant to 35 U.S.C. §§ 284-85; and

E. For such other relief as the Court may deem just and fair.

DEMAND FOR JURY TRIAL

Pursuant to Fed. R. Civ. P. 38, plaintiff demands a jury trial as to all matters triable of right by a jury.

DATED: May 15, 2009

CORR CRONIN MICHELSON
BAUMGARDNER & PREECE LLP

s/William F. Cronin
William F. Cronin, WSBA No. 8667

CORR CRONIN MICHELSON
BAUMGARDNER & PREECE LLP
1001 Fourth Avenue, Suite 3900
Seattle, WA 98154
Telephone: (206) 625-8600
Facsimile: (206) 625-0900
wcronin@corrcronin.com

JOSH A. KREVITT*
Y. ERNEST HSIN*
SARAH E. PIEPMEIER*
GIBSON, DUNN & CRUTCHER LLP
200 Park Avenue
New York, NY 10166-0193
Telephone: (212) 351-4000
Facsimile: (212) 351-4035
*(*pro hac vice applications pending)*

Attorneys for Plaintiff AMAZON.COM, INC.

EXHIBIT 1

US006006225A

United States Patent [19]

Bowman et al.

[11] **Patent Number: 6,006,225**

[45] **Date of Patent: Dec. 21, 1999**

[54] **REFINING SEARCH QUERIES BY THE SUGGESTION OF CORRELATED TERMS FROM PRIOR SEARCHES**

[75] Inventors: **Dwayne E. Bowman**, Woodinville; **Ruben E. Ortega**; **Michael L. Hamrick**, both of Seattle; **Joel R. Spiegel**, Woodinville; **Timothy R. Kohn**, Seattle, all of Wash.

[73] Assignee: **Amazon.Com**, Seattle, Wash.

[21] Appl. No.: **09/145,360**

[22] Filed: **Sep. 1, 1998**

Related U.S. Application Data

[60] Provisional application No. 60/089,244, Jun. 15, 1998.

[51] **Int. Cl.**[6] **G06F 17/30**

[52] **U.S. Cl.** **707/5**; 707/2; 707/4; 707/10

[58] **Field of Search** 707/5, 2, 10, 4

[56] **References Cited**

U.S. PATENT DOCUMENTS

5,675,819	10/1997	Schuetze	704/10
5,721,897	2/1998	Rubinstein	707/2
5,787,422	7/1998	Tukey et al.	707/5
5,794,233	8/1998	Rubinstein	707/4
5,864,845	1/1999	Voorhees et al.	707/5
5,911,140	6/1999	Tukey et al.	707/5
5,913,215	6/1999	Rubinstein	707/10

OTHER PUBLICATIONS

Bartell et al., "Automatic Combination of Multiple Ranked Retrieval Systems", Proceedings of SIGIR '94, Jul. 1994, pp. 173–181.

Belkin et al., "The Effect of Multiple Query Representations on Information System Performance" Proceedings of SIGIR '93, Jun. 1993, pp. 339–346.

Shaw et al., "Combination of Multiple Searches", Proceedings of TREC–3, Apr. 1995, pp. 105–108.

QuarterDeck Web Page, Downloaded Sep. 9, 1996, http://aracnid.qdeck.com/qdeck/products/webcompass.

Towell et al. "Learning Collection Fusion Strategies for Information Retrieval", Proceedings of the 12th Annual Machine Learning Conference, Jul. 1995, pp. 540–548.

Voorhees et al., "Learning Collection Fusion Strategies", Proceedings of SIGIR '95, Jul. 1995, pp. 172–179.

Voorhees et al., "The Collection Fusion Problem" Proceedings of TREC–3, NIST Special Publication 500–225, Apr. 1995, pp. 95–104.

Abstract of *Generating Advanced Query Interfaces*, Lee, Srivastava and Vista, Computer Networks and ISDN Systems Conference Title: Comput. Netw. ISDN Syst. (Netherlands) vol. 30, No. 1–7, pp. 656–657 (1998).

Abstract of *Using Combination of Evidence for Term Expansion*, Wilkinson, Information Retrieval Research, Proceedings of the 19th Annual BCS–IRSG Colloquium on IR Research (1997).

(List continued on next page.)

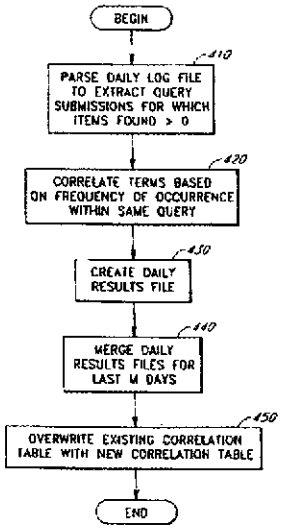
Primary Examiner—Paul R. Lintz

Attorney, Agent, or Firm—Knobbe, Martens, Olson & Bear, LLP

[57] **ABSTRACT**

A search engine is disclosed which suggests related terms to the user to allow the user to refine a search. The related terms are generated using query term correlation data which reflects the frequencies with which specific terms have previously appeared within the same query. The correlation data is generated and stored in a look-up table using an off-line process which parses a query log file. The table is regenerated periodically from the most recent query submissions (e.g., the last two weeks of query submissions), and thus strongly reflects the current preferences of users. Each related term is presented to the user via a respective hyperlink which can be selected by the user to submit a modified query. In one embodiment, the related terms are added to and selected from the table so as to guarantee that the modified queries will not produce a NULL query result.

28 Claims, 10 Drawing Sheets



OTHER PUBLICATIONS

Abstract of *Inquirus, the NECI Meta Search Engine*, Lawrence and Giles, Computer Networks and ISDN Systems Conference Title: Comput. Netw ISDN Syst. (Netherlands) vol. 30, No. 1–7, pp. 95–105 (1998).

Abstract of *Facilitating Complex Web Queries Through Visual User Interfaces and Query Relaxtion*, Li and Shim, Computer Networks and ISDN Systems Conference Title: Comput. Netw. ISDN Syst. (Netherlands) vol. 30, No. 1–7, pp. 149–159 (1998).

A User–centred Evaluation of Ranking Algorithms for Interactive Query Expansion, Efthimiadis, Proceedings of the 16th Annual International ACM SIGIR Conference, Pittsburgh, pp. 146–159 (1993).

Concept Based Query Expansion, Qiu and Frei, Proceedings of the 16th Annual International ACM SIGIR Conference, Pittsburgh, pp. 160–169 (1993)

Improving Retrieval Performance by Relevance Feedback, Salton and Buckley, J. of Am. Society for Info. Science 41(4):288–297 (1990).

Query Expansion Using Domain–Adapted, Weighted Thesaurus in an Extended Boolean Model, Kwon, Kim and Choi, Proceedings of the 3rd International Conference on Information and Knowledge Management (CIKM'94), pp. 140–146 (1994).

Browsing Through Querying: Designing for Electronic Books, Charoenkitkarn, Tam, Chignell and Golovichinsky, at the 5th ACM Conference on Hypertext, Seattle, WA 206–216 (1993).

A Survey of Information Retrieval and Filtering Methods, Faloutsos and Oard, Univ. of Maryland, 22 pages (undated).

A Corpus Analysis Approach for Automatic Query Expansion, Gauch and Wang, Proceedings of the 6th International Conference on Information and Knowledge Mangement, pp. 278–284 (1997).

Discovering Web Acess Patterns and Trends by Applying OLAP and Data Mining Technology on Web Logs, Zaiane, Xin and Han, Proceedings of the IEEE Forum on Research and Technology Advances in Digital Libraries (IEEE ADL'98), pp. 19–29 (1998).

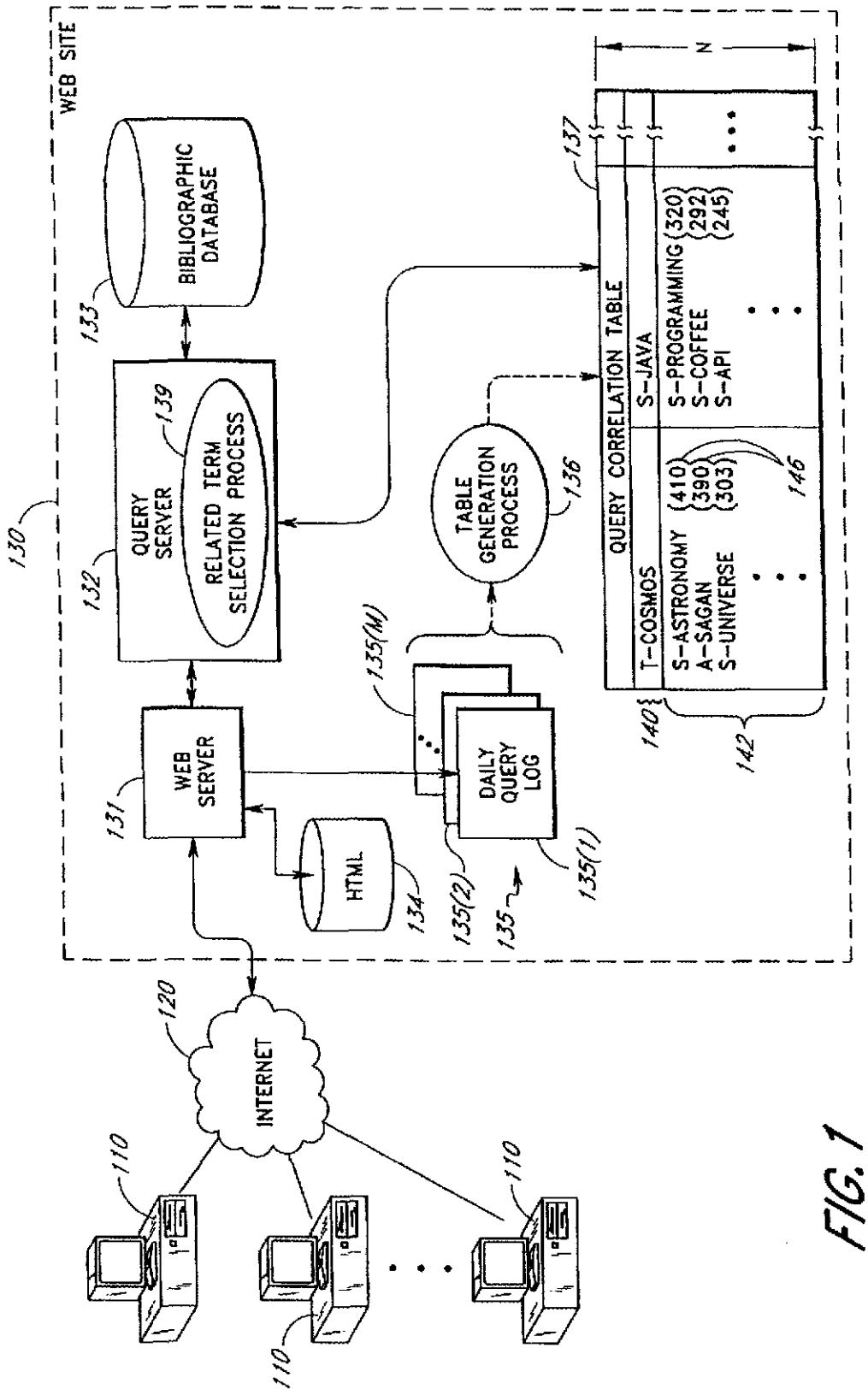

WEB SITE
130
BIBLIOGRAPHIC DATABASE
133
QUERY SERVER
132
RELATED TERM SELECTION PROCESS
139
TABLE GENERATION PROCESS
136
WEB SERVER
131
HTML
134
DAILY QUERY LOG
135(1)
135(2)
135(M)
135
INTERNET
120
110
110
110
QUERY CORRELATION TABLE
137
T-COSMOS
S-ASTRONOMY (410)
A-SAGAN (390)
S-UNIVERSE (303)
S-JAVA
S-PROGRAMMING (320)
S-COFFEE (292)
S-API (245)
N
140
142
146

FIG. 1

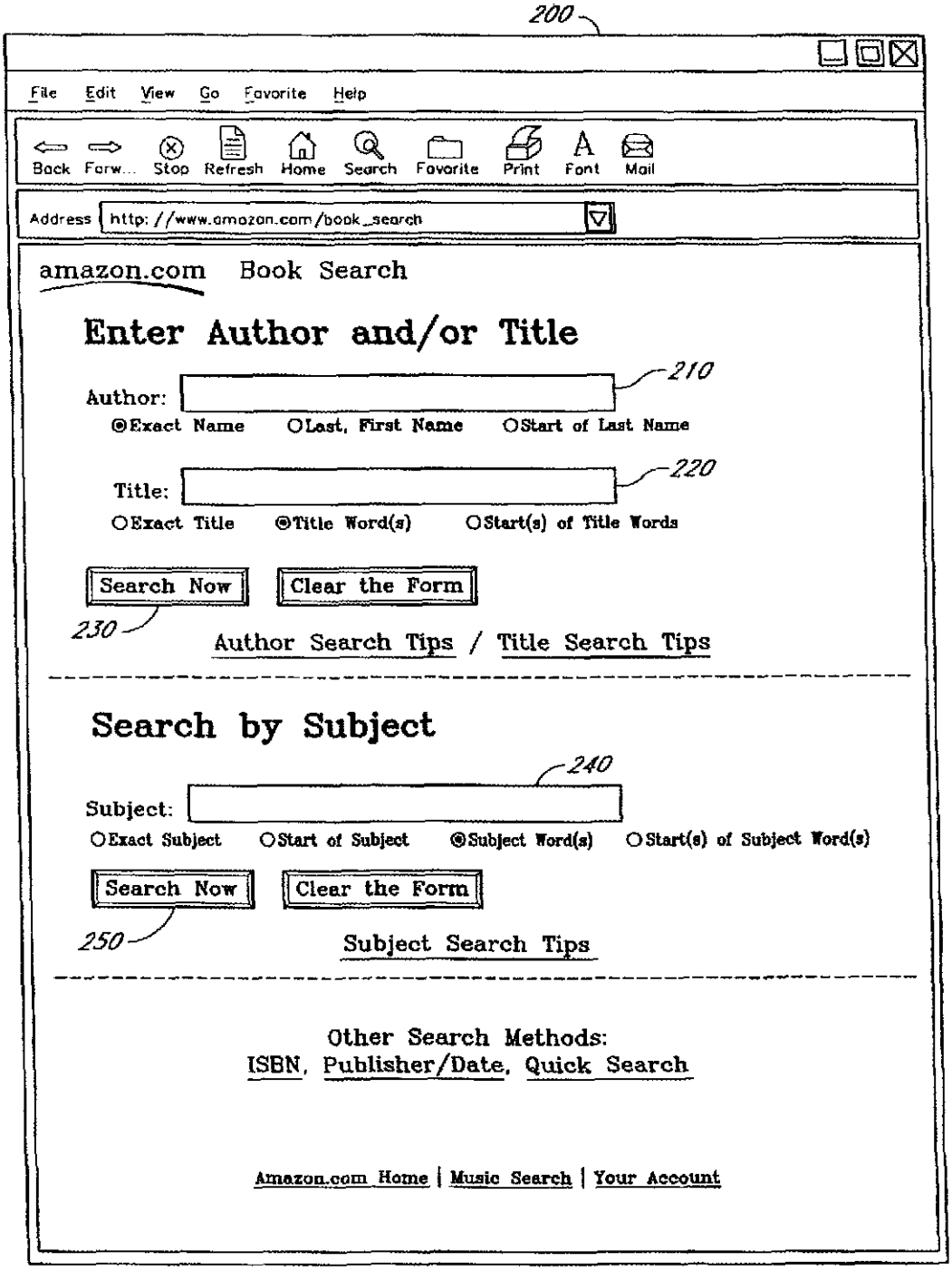

200
File Edit View Go Favorite Help
Back Forw... Stop Refresh Home Search Favorite Print Font Mail
Address http://www.amazon.com/book_search
amazon.com Book Search
Enter Author and/or Title
Author:
210
Exact Name Last, First Name Start of Last Name
Title:
220
Exact Title Title Word(s) Start(s) of Title Words
Search Now
Clear the Form
230
Author Search Tips / Title Search Tips
Search by Subject
240
Subject:
Exact Subject Start of Subject Subject Word(s) Start(s) of Subject Word(s)
Search Now
Clear the Form
250
Subject Search Tips
Other Search Methods:
ISBN, Publisher/Date, Quick Search
Amazon.com Home | Music Search | Your Account

FIG. 2

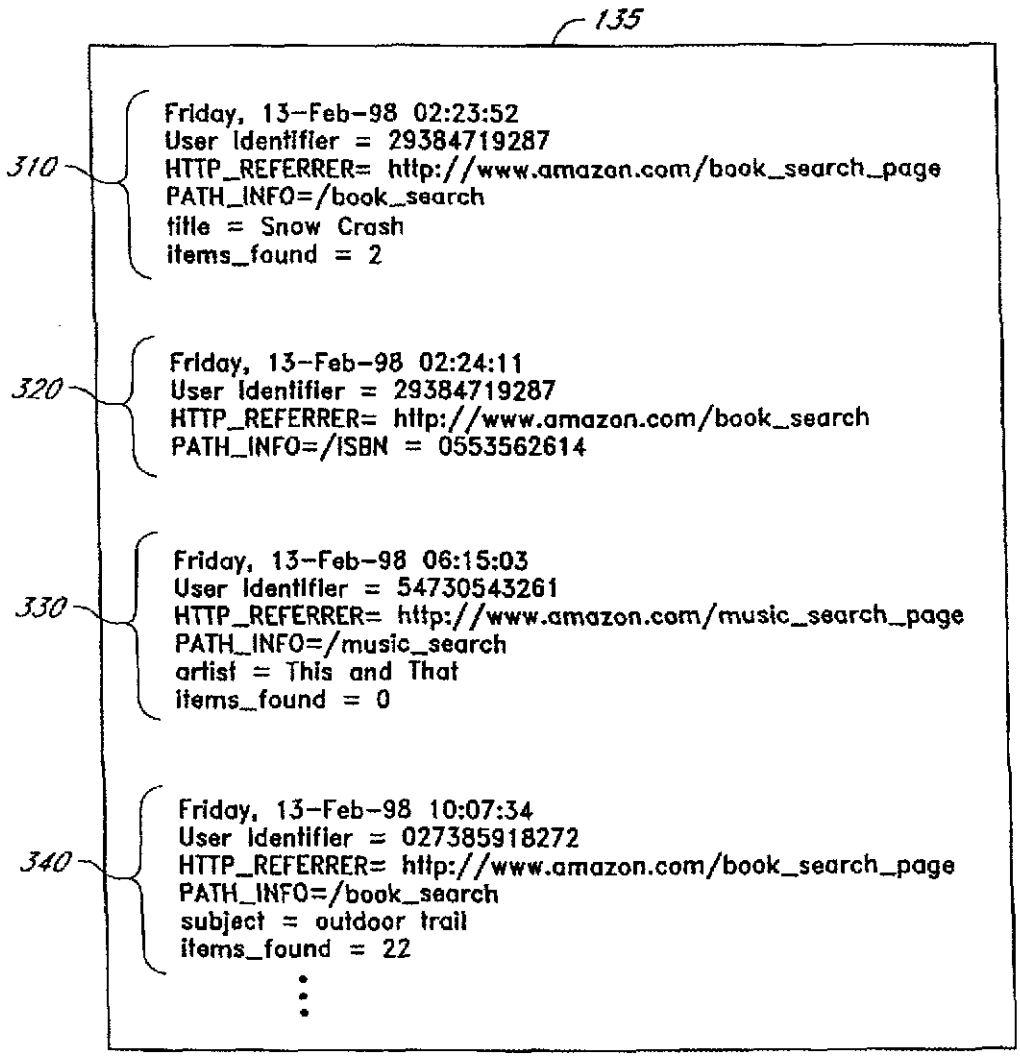

135
310
Friday, 13-Feb-98 02:23:52
User Identifier = 29384719287
HTTP_REFERRER= http://www.amazon.com/book_search_page
PATH_INFO=/book_search
title = Snow Crash
items_found = 2
320
Friday, 13-Feb-98 02:24:11
User Identifier = 29384719287
HTTP_REFERRER= http://www.amazon.com/book_search
PATH_INFO=/ISBN = 0553562614
330
Friday, 13-Feb-98 06:15:03
User Identifier = 54730543261
HTTP_REFERRER= http://www.amazon.com/music_search_page
PATH_INFO=/music_search
artist = This and That
items_found = 0
340
Friday, 13-Feb-98 10:07:34
User Identifier = 027385918272
HTTP_REFERRER= http://www.amazon.com/book_search_page
PATH_INFO=/book_search
subject = outdoor trail
items_found = 22

FIG. 3

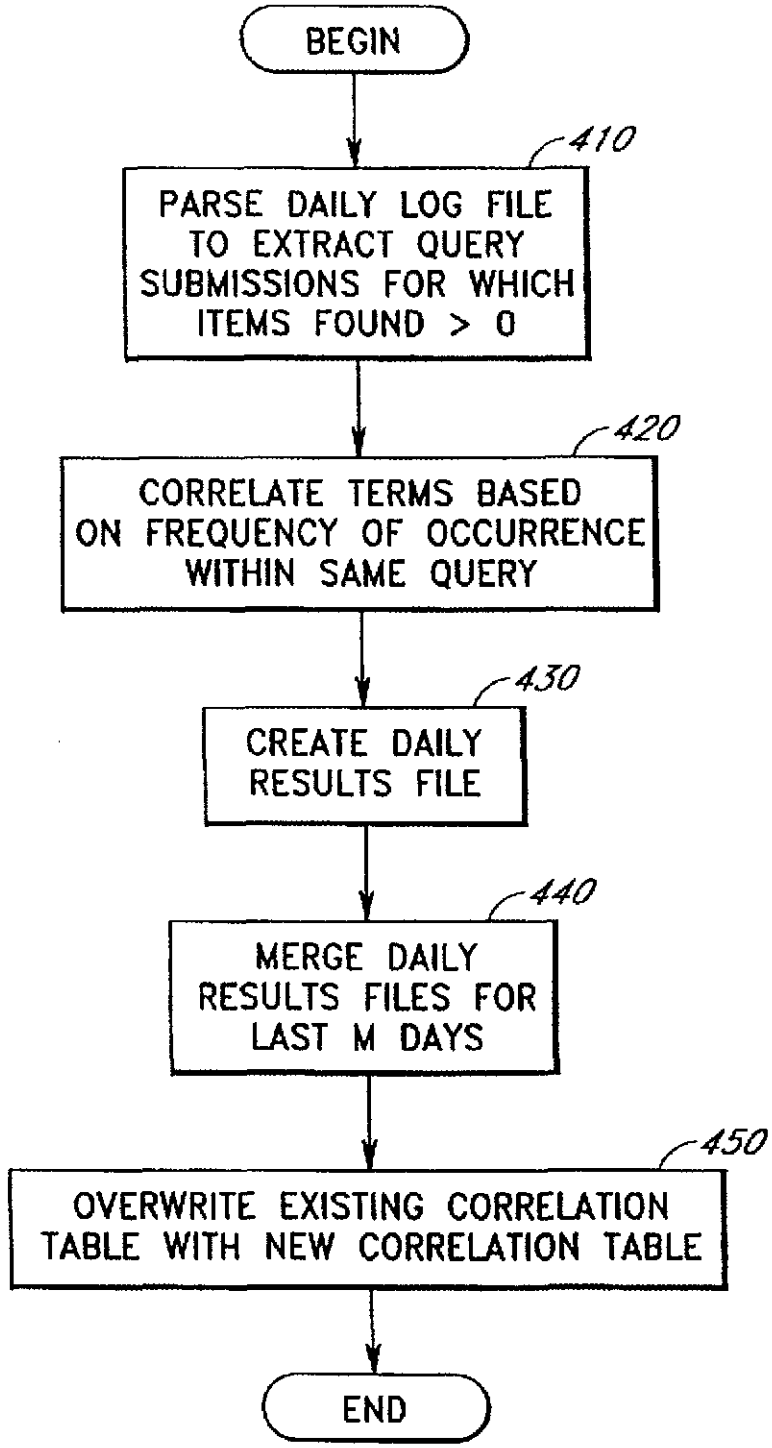

BEGIN
410
PARSE DAILY LOG FILE TO EXTRACT QUERY SUBMISSIONS FOR WHICH ITEMS FOUND > 0
420
CORRELATE TERMS BASED ON FREQUENCY OF OCCURRENCE WITHIN SAME QUERY
430
CREATE DAILY RESULTS FILE
440
MERGE DAILY RESULTS FILES FOR LAST M DAYS
450
OVERWRITE EXISTING CORRELATION TABLE WITH NEW CORRELATION TABLE
END

FIG. 4

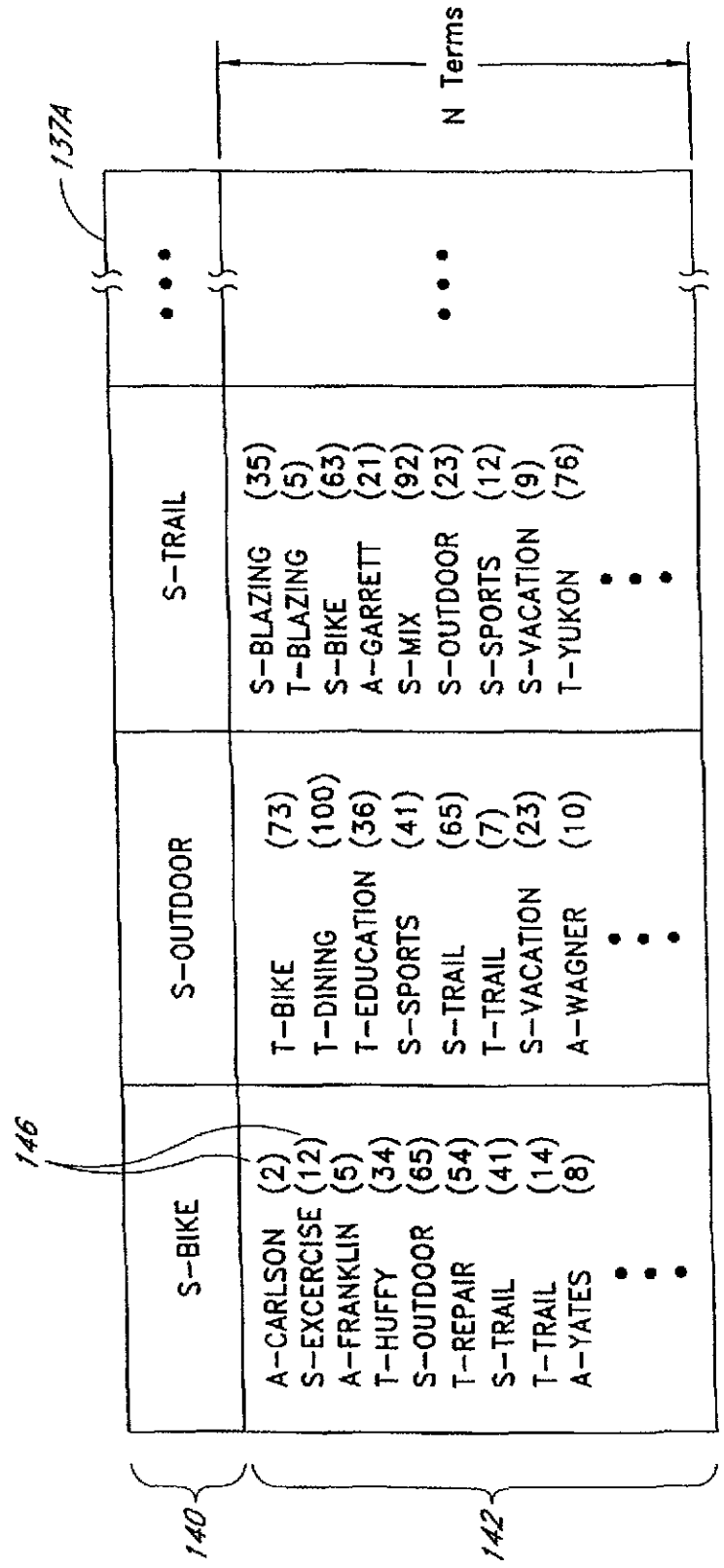

137A
N Terms
S-BIKE
S-OUTDOOR
S-TRAIL
A-CARLSON (2)
S-EXCERCISE (12)
A-FRANKLIN (5)
T-HUFFY (34)
S-OUTDOOR (65)
T-REPAIR (54)
S-TRAIL (41)
T-TRAIL (14)
A-YATES (8)
T-BIKE (73)
T-DINING (100)
T-EDUCATION (36)
S-SPORTS (41)
S-TRAIL (65)
T-TRAIL (7)
S-VACATION (23)
A-WAGNER (10)
S-BLAZING (35)
T-BLAZING (5)
S-BIKE (63)
A-GARRETT (21)
S-MIX (92)
S-OUTDOOR (23)
S-SPORTS (12)
S-VACATION (9)
T-YUKON (76)
146
140
142

FIG. 5A

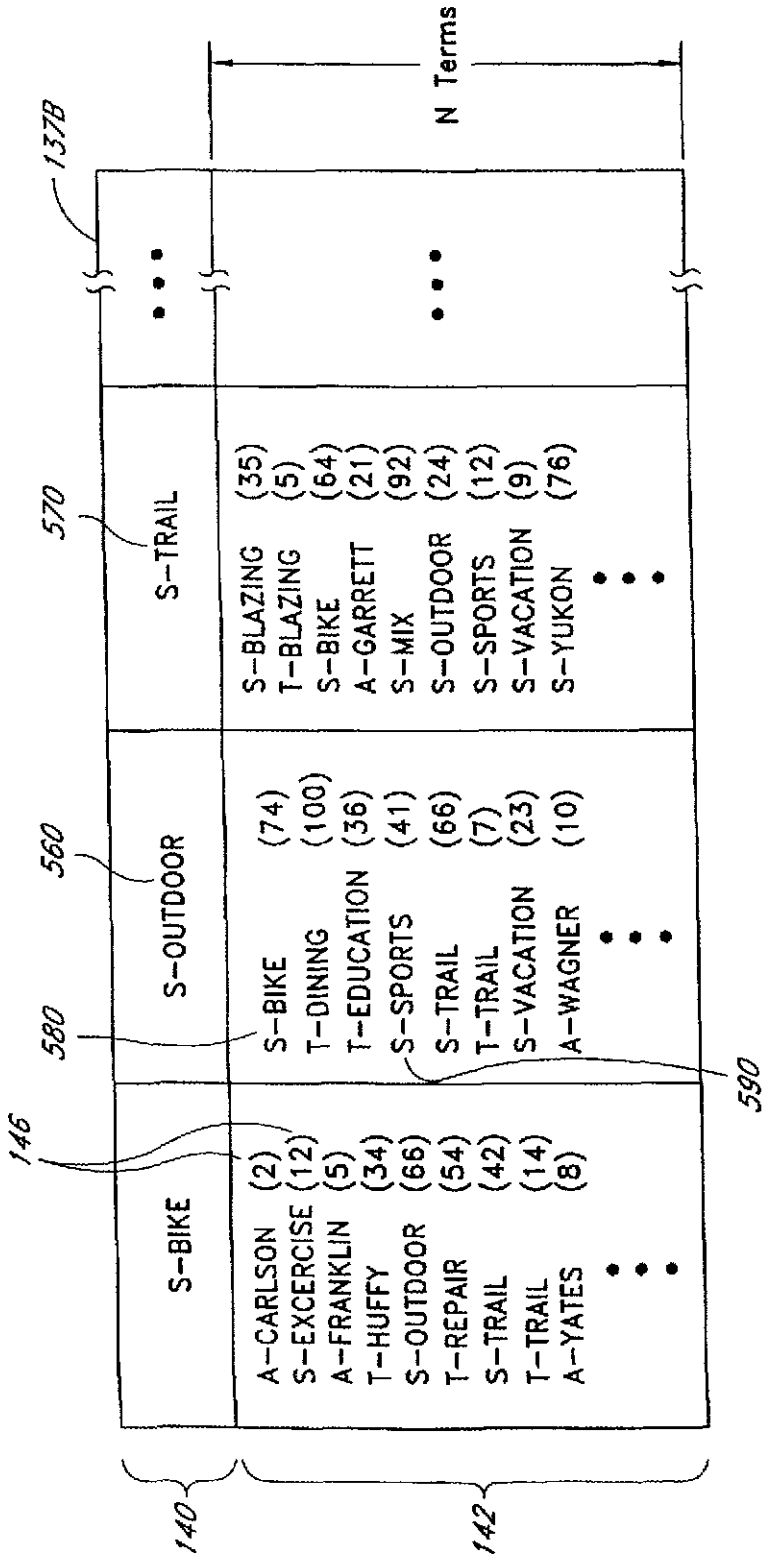

N Terms
137B
570
560
580
590
146
140
142
S-BIKE
S-OUTDOOR
S-TRAIL
A-CARLSON (2)
S-EXCERCISE (12)
A-FRANKLIN (5)
T-HUFFY (34)
S-OUTDOOR (66)
T-REPAIR (54)
S-TRAIL (42)
T-TRAIL (14)
A-YATES (8)
S-BIKE (74)
T-DINING (100)
T-EDUCATION (36)
S-SPORTS (41)
S-TRAIL (66)
T-TRAIL (7)
S-VACATION (23)
A-WAGNER (10)
S-BLAZING (35)
T-BLAZING (5)
S-BIKE (64)
A-GARRETT (21)
S-MIX (92)
S-OUTDOOR (24)
S-SPORTS (12)
S-VACATION (9)
S-YUKON (76)

FIG. 5B

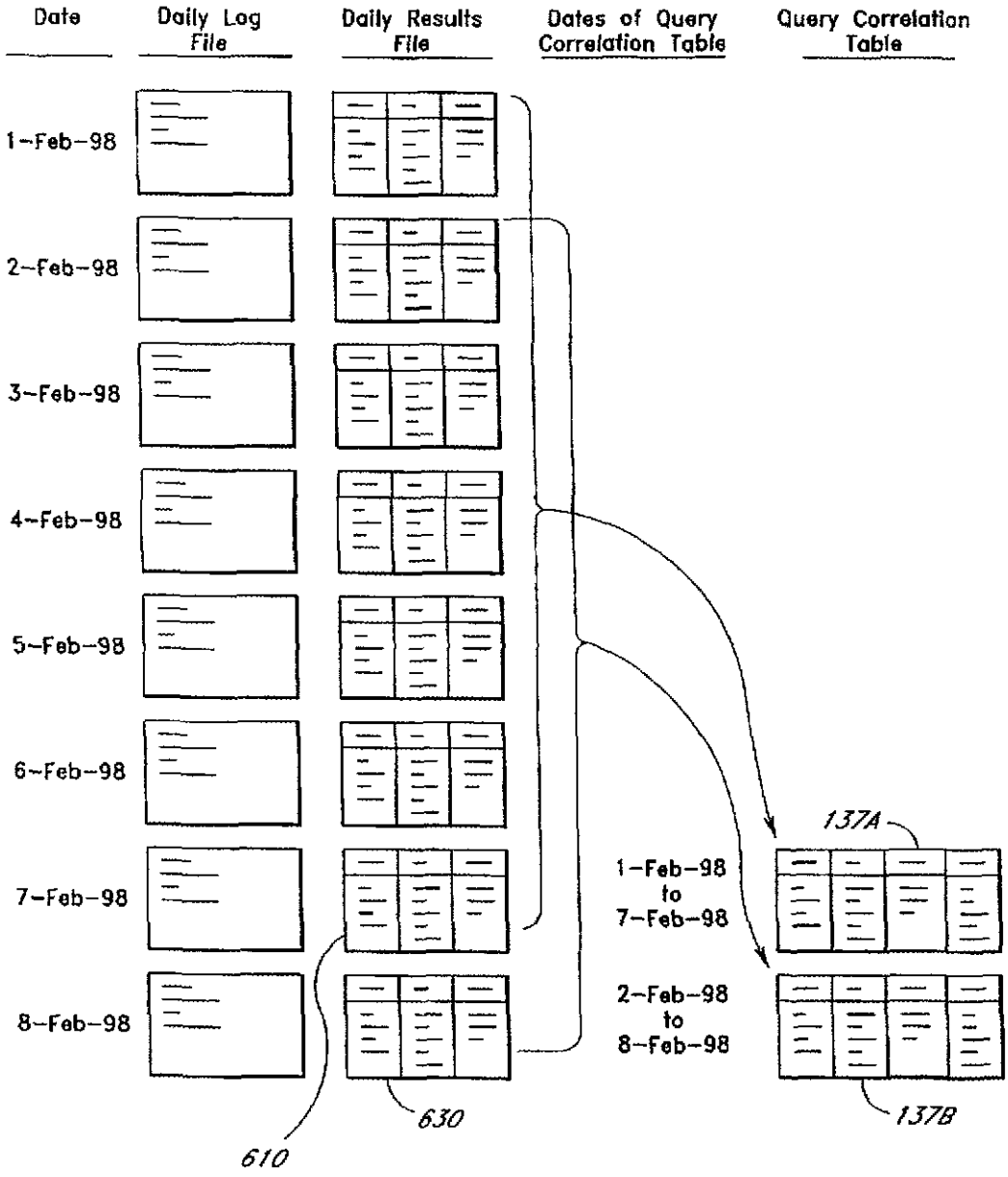

Date
Daily Log File
Daily Results File
Dates of Query Correlation Table
Query Correlation Table
1-Feb-98
2-Feb-98
3-Feb-98
4-Feb-98
5-Feb-98
6-Feb-98
7-Feb-98
8-Feb-98
1-Feb-98 to 7-Feb-98
2-Feb-98 to 8-Feb-98
137A
137B
610
630

FIG. 6

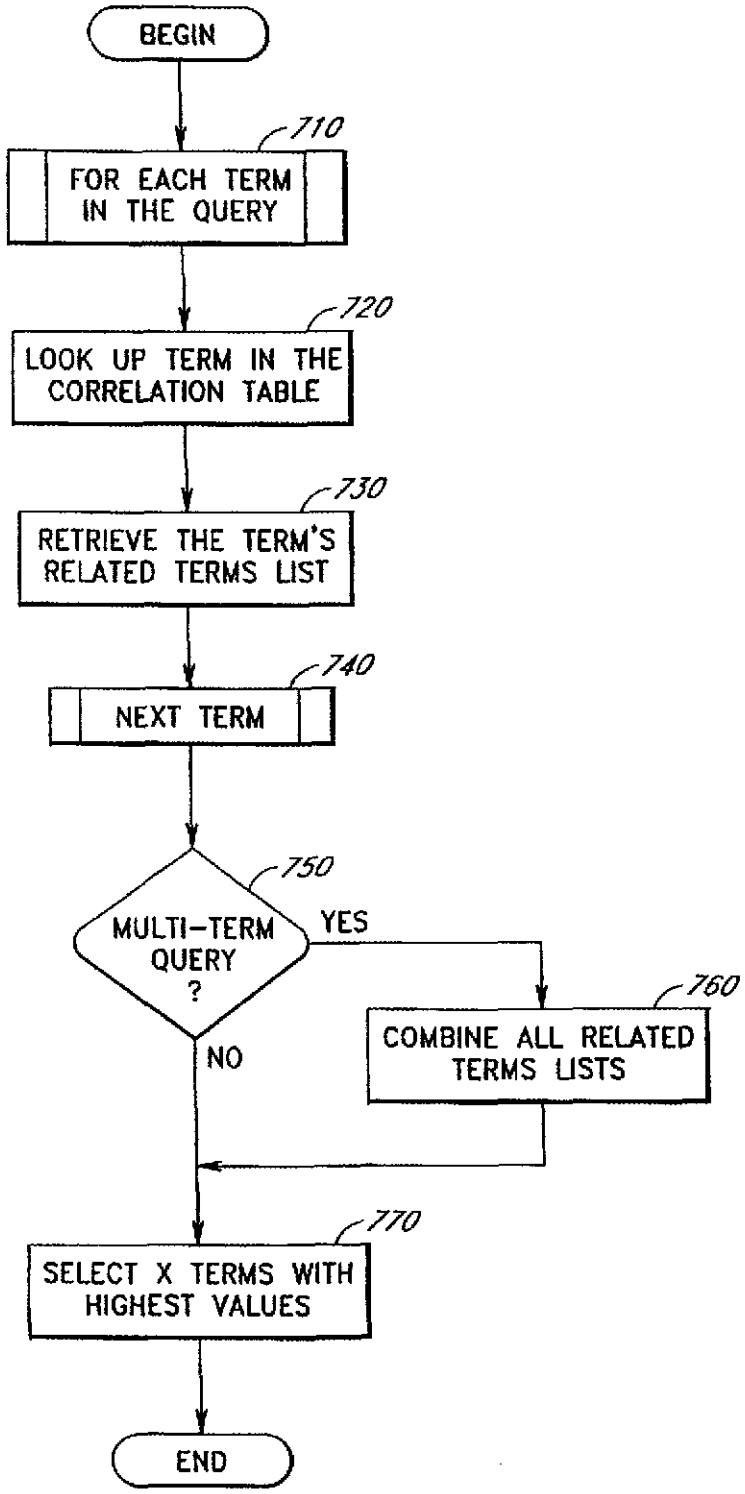

BEGIN
710
FOR EACH TERM IN THE QUERY
720
LOOK UP TERM IN THE CORRELATION TABLE
730
RETRIEVE THE TERM'S RELATED TERMS LIST
740
NEXT TERM
750
MULTI-TERM QUERY ?
YES
NO
760
COMBINE ALL RELATED TERMS LISTS
770
SELECT X TERMS WITH HIGHEST VALUES
END

FIG. 7

Top 3 Related Terms:

810

TRAIL – MIX
TRAIL – YUKON
TRAIL – BIKE

FIG. 8A

Intersecting Terms:

820

S – BIKE
S – SPORTS
S – VACATION

830

Top 3 Related Terms:

OUTDOOR TRAIL – BIKE
OUTDOOR TRAIL – SPORTS
OUTDOOR TRAIL – VACATION

FIG. 8B

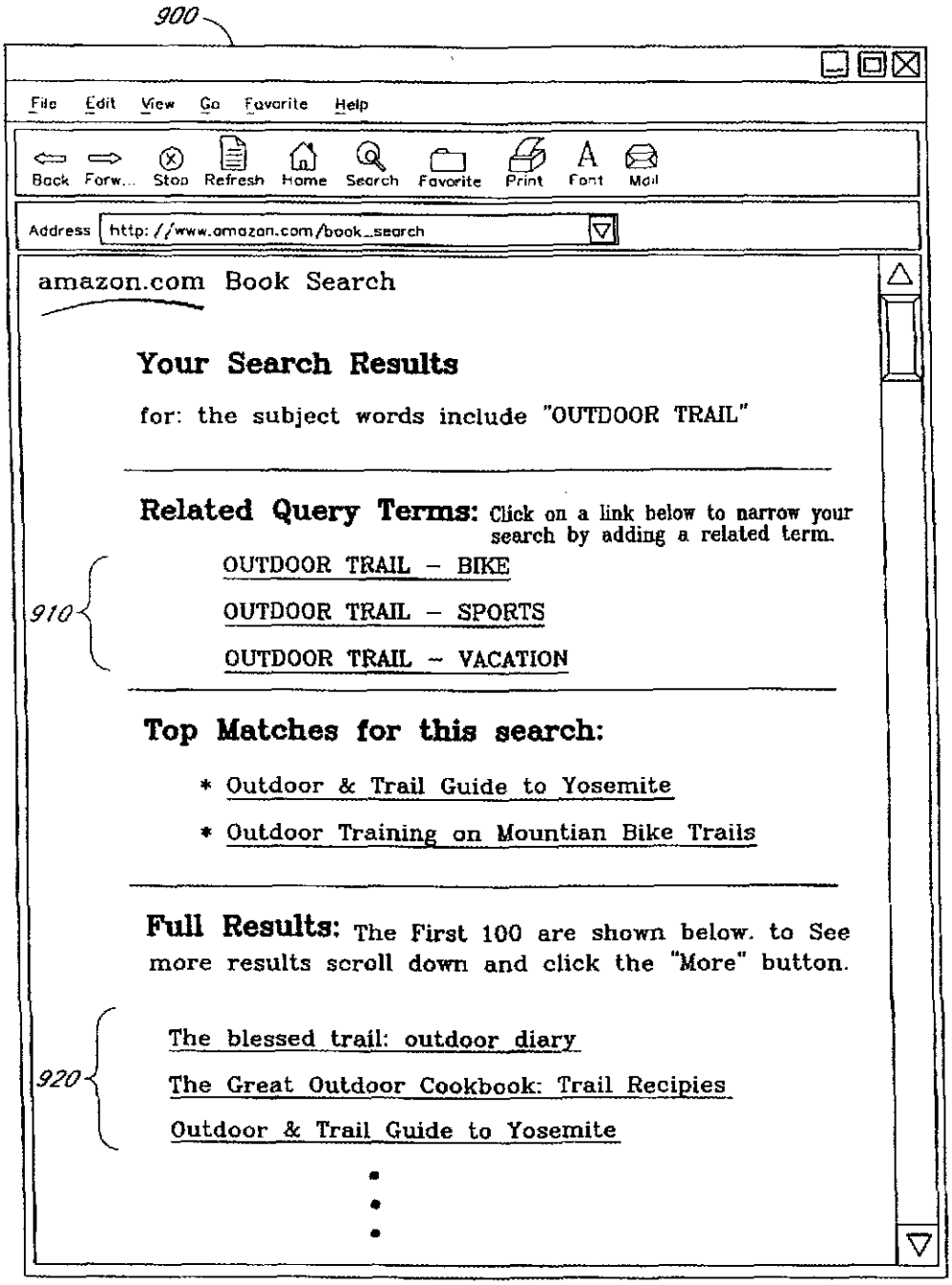

900
File Edit View Go Favorite Help
Back Forw... Stop Refresh Home Search Favorite Print Font Mail
Address http://www.amazon.com/book_search
amazon.com Book Search
Your Search Results
for: the subject words include "OUTDOOR TRAIL"
Related Query Terms: Click on a link below to narrow your search by adding a related term.
910
OUTDOOR TRAIL - BIKE
OUTDOOR TRAIL - SPORTS
OUTDOOR TRAIL - VACATION
Top Matches for this search:
* Outdoor & Trail Guide to Yosemite
* Outdoor Training on Mountian Bike Trails
Full Results: The First 100 are shown below. to See more results scroll down and click the "More" button.
920
The blessed trail: outdoor diary
The Great Outdoor Cookbook: Trail Recipies
Outdoor & Trail Guide to Yosemite

FIG. 9

REFINING SEARCH QUERIES BY THE SUGGESTION OF CORRELATED TERMS FROM PRIOR SEARCHES

RELATED APPLICATION

This application claims the benefit of U. S. Provisional Application No. 60/089,244, filed Jun. 15, 1998, the disclosure of which is hereby incorporated by reference.

BACKGROUND OF THE INVENTION

1. Field of Invention

This present invention relates to query processing, and more specifically relates to techniques for facilitating the process of refining search queries.

2. Description of Related Art

With the increasing popularity of the Internet and the World Wide Web, it is common for on-line users to utilize search engines to search the Internet for desired information. Many web sites permit users to perform searches to identify a small number of relevant items among a much larger domain of items. As an example, several web index sites permit users to search for particular web sites among known web sites. Similarly, many on-line merchants, such as booksellers, permit users to search for particular products among all of the products that can be purchased from the merchant. Other on-line services, such as Lexis™ and Westlaw™, allow users to search for various articles and court opinions.

In order to perform a search, a user submits a query containing one or more query terms. The query may also explicitly or implicitly identify a record field or segment to be searched, such as title, author, or subject classification of the item. For example, a user of an on-line bookstore may submit a query containing terms that the user believes appear within the title of a book. A query server program of the search engine processes the query to identify any items that match the terms of the query. The set of items identified by the query server program is referred to as a "query result." In the on-line bookstore example, the query result is a set of books whose titles contain some or all of the query terms. In the web index site example, the query result is a set of web sites or documents. In web-based implementations, the query result is typically presented to the user as a hypertextual listing of the located items.

If the scope of the search is large, the query result may contain hundreds, thousands or even millions of items. If the user is performing the search in order to find a single item or a small set of items, conventional approaches to ordering the items within the query result often fail to place the sought item or items near the top of the query result list. This requires the user to read through many other items in the query result before reaching the sought item. Certain search engines, such as Excite™ and AltaVista™, suggest related query terms to the user as a part of the "search refinement" process. This allows the user to further refine the query and narrow the query result by selecting one or more related query terms that more accurately reflect the user's intended request. The related query terms are typically generated by the search engine using the contents of the query result, such as by identifying the most frequently used terms within the located documents. For example, if a user were to submit a query on the term "FOOD," the user may receive several thousand items in the query result. The search engine might then trace through the contents of some or all of these items and present the user with related query terms such as "RESTAURANTS," "RECIPIES," and "FDA" to allow the user to refine the query.

The related query terms are commonly presented to the user together with corresponding check boxes that can be selectively marked or checked by the user to add terms to the query. In some implementations, the related query terms are alternatively presented to and selected by the user through drop down menus that are provided on the query result page. In either case, the user can add additional terms to the query and then resubmit the modified query. Using this technique, the user can narrow the query result down to a more manageable set consisting primarily of relevant items.

One problem with existing techniques for generating related query terms is that the related terms are frequently of little or no value to the search refinement process. Another problem is that the addition of one or more related terms to the query sometimes leads to a NULL query result. Another problem is that the process of parsing the query result items to identify frequently used terms consumes significant processor resources, and can appreciably increase the amount of time the user must wait before viewing the query result. These and other deficiencies in existing techniques hinder the user's goal of quickly and efficiently locating the most relevant items, and can lead to user frustration.

SUMMARY OF THE INVENTION

The present invention addresses these and other problems by providing a search refinement system and method for generating and displaying related query terms ("related terms"). In accordance with the invention, the related terms are generating using query term correlation data that is based on historical query submissions to the search engine. The query term correlation data ("correlation data") is preferably based at least upon the frequencies with which specific terms have historically been submitted together within the same query. The incorporation of such historical query information into the process tends to produce related terms that are frequently used by other users in combination with the submitted query terms, and significantly increases the likelihood that these related terms will be helpful to the search refinement process. To further increase the likelihood that the related terms will be helpful, the correlation data is preferably generated only from those historical query submissions that produced a successful query result (at least one match).

In accordance with one aspect of the invention, the correlation data is stored in a correlation data structure (table, database, etc.) which is used to look up related terms in response to query submissions. The data structure is preferably generated using an off-line process which parses a query log file, but could alternatively be generated and updated in real-time as queries are received from users. In one embodiment, the data structure is regenerated periodically (e.g., once per day) from the most recent query submissions (e.g., the last M days of entries in the query log), and thus strongly reflects the current tastes of the community of users; as a result, the related terms suggested by the search engine strongly reflect the current tastes of the community. Thus, for example, in the context of a search engine of an online merchant, the search engine tends to suggest related terms that correspond to the current best-selling products.

In a preferred embodiment, each entry in the data structure is in the form of a key term and a corresponding related terms list. Each related terms list contains the terms which have historically appeared together (in the same query) with

the respective key term with the highest degree of frequency, ignoring unsuccessful query submissions (query submissions which produced a NULL query result). The data structure thus provides an efficient mechanism for looking up the related terms for a given query term.

To generate a set of related terms for refining a submitted query (the "present query"), the related terms list for each term in the present query is initially obtained from the correlation data structure. If this step produces multiple related terms lists (as in the case of a multiple-term query), the related terms lists are preferably combined by taking the intersection between these lists (i.e., deleting the terms that are not common to all lists). The related terms which remain are terms which have previously appeared, in at least one successful query submission, in combination with every term of the present query. Thus, assuming items have not been deleted from the database being searched, any of these related terms can be individually added to the present query while guaranteeing that the modified query will not produce a NULL query result. To take advantage of this feature, the related terms are preferably presented to the user via a user interface that requires the user to add no more than one related term per query submission. In other embodiment, the related terms are selected and displayed without guaranteeing a successful query result.

Because the related terms are identified from previously-generated correlation data, without the need to parse documents or correlate terms, the related terms can be identified and presented to the user with little or no added delay.

BRIEF DESCRIPTION OF THE DRAWINGS

These and other features will now be described with reference to the drawings summarized below. These drawings and the associated description are provided to illustrate a preferred embodiment of the invention, and not to limit the scope of the invention.

Throughout the drawings, reference numbers are re-used to indicate correspondence between referenced elements. In addition, the first digit of each reference number indicates the figure in which the element first appears.

FIG. **1** illustrates a system in which users access web site information via the Internet, and illustrates the basic web site components used to implement a search engine which operates in accordance with the invention.

FIG. **2** illustrates a sample book search page of the web site.

FIG. **3** illustrates sample log entries of a daily query log file.

FIG. **4** illustrates the process used to generate the correlation table of FIG. **1**.

FIG. **5**A illustrates a sample mapping before a query is added.

FIG. **5**B illustrates a sample mapping after a query is added.

FIG. **6** illustrates a process for generating the correlation table from the most recent daily query log files.

FIG. **7** illustrates a process for selecting the related query terms from the correlation table.

FIG. **8**A illustrates a set of related query terms from a single-term query.

FIG. **8**B illustrates a set of intersecting terms and a set of related query terms from a multiple-term query.

FIG. **9** illustrates a sample search result page of the web site.

DETAILED DESCRIPTION OF THE PREFERRED EMBODIMENTS

The present invention provides a search refinement system and method for generating related query terms ("related terms") using a history of queries submitted to a search engine by a community of users. Briefly, the system generates query term correlation data which reflects the frequency with which specific terms have previously occurred together within the same query. The system uses the query term correlation data in combination with the query term(s) entered by the user to recommend additional query terms for refining the query. The incorporation of such historical query information into the process tends to produce related terms that are frequently used by other users in combination with the submitted query terms, and significantly increases the likelihood that these related terms will be helpful to the search refinement process. To further increase the likelihood that the related terms will be helpful, the correlation data is preferably generated only from those historical query submissions that produced a successful query result (at least one match).

In the preferred embodiment, the query term correlation date is regenerated periodically from recent query submissions, such as by using the last M days of entries in a query log, and thus heavily reflects the current tastes of users. As a result, the related terms suggested by the search engine tend to be terms that correspond to the most frequently searched items during the relevant time period. Thus, for example, in the context of a search engine of an online merchant, the search engine tends to suggest related terms that correspond to the current best-selling products. In one embodiment, the technique used to generate the related terms and present these terms to the user guarantees that the modified query will not produce a NULL query result.

The search refinement methods of the invention may be implemented, for example, as part of a web site, an Internet site, an on-line services network, a document retrieval system, or any other type of computer system that provides searching capabilities to a community of users. In addition, the method may be combined with other methods for suggesting related terms, such as methods which process the contents of located documents.

A preferred web-based implementation of the search refinement system will now be described with reference to FIGS. **1**–**9**. For purposes of illustration, the system is described herein in the context of a search engine that is used to assist customers of Amazon.com Inc. in locating items (e.g., books, CDs, etc.) from an on-line catalog of products. Throughout the description, reference will be made to various implementation-specific details of the Amazon.com implementation. These details are provided in order to fully illustrate a preferred embodiment of the invention, and not to limit the scope of the invention. The scope of the invention is set forth in the appended claims.

I. Overview of Web Site and Search Engine

FIG. **1** illustrates the Amazon.com web site **130**, including components used to implement a search engine in accordance with the invention.

As it is well known in the art of Internet commerce, the Amazon.com web site includes functionality for allowing users to search, browse, and make purchases from an online catalog of book titles, music titles, and other types of items via the Internet **120**. Because the catalog contains millions of items, it is important that the site provide an efficient mechanism for assisting users in locating items.

As illustrated by FIG. 1, the web site 130 includes a web server application 131 ("web server") which processes user requests received from user computers 110 via the Internet 120. These requests include queries submitted by users to search the on-line catalog for products. The web server 131 records the user transactions, including query submissions, within a query log 135. In the embodiment depicted in FIG. 1, the query log 135 consists of a sequence of daily query log files 135(1)–135(M), each of which represents one day of transactions.

The web site 130 also includes a query server 132 which processes the queries by searching a bibliographic database 133. The bibliographic database 133 includes information about the various products that users may purchase through the web site 130. This information includes, for example, the titles, authors, publishers, subject descriptions, and ISBNs (International Standard Book Numbers) of book titles, and the titles, artists, labels, and music classifications of music titles. The information for each item is arranged within fields (such as an "author" field and a "title" field), enabling the bibliographic database 133 to be searched on a field-restricted basis. The site also includes a database 134 of HTML (Hypertext Markup Language) content which includes, among other things, product information pages which show and describe the various products.

The query server 132 includes a related term selection process 139 which identifies related query terms based on query term correlation data stored in a correlation table 137. As depicted in FIG. 1 and described below, the correlation table 137 is generated periodically from the M most recent daily query log files 135(1)–135(M) using an off-line table generation process 136.

The web server 131, query server 132, table generation process 136, and database software run on one or more Unix™-based servers and workstations (not shown) of the web site 130 although other types of platforms could be used. The correlation table 137 is preferably cached in RAM (random access memory) on the same physical machine as that used to implement the query server 132. To accommodate large numbers of users, they query server 132 and the correlation table 137 can be replicated across multiple machines. The web site components that are invoked during the searching process are collectively referred to herein as a "search engine."

FIG. 2 illustrates the general format of a book search page 200 of the Amazon.com web site 130 that can be used to search the bibliographic database 133 for book titles. Users have access to other search pages that can be used to locate music titles and other types of products sold by the on-line merchant. The book search page 200 includes author, title, and subject fields 210, 220, 240 and associated controls that allow the user to initiate fieldrestricted searches for book titles. Users can perform searches by first typing in the desired information into a search field 210, 220, 240 and then clicking on the appropriate search button 230, 250. The term or string of terms submitted to the search engine is referred to herein as the "query." Other areas of the web site ask the user to submit queries without limiting the terms to specific fields.

When the user submits a query from the book search page 200 to the web site 130, a the query sever 132 applies the query to the bibliographic database 133, taking into account any field restrictions within the query. If the query result is a single item, the item's product information page is presented to the user. If the query result includes multiple items, the list of items is presented to the user through a query result page which contains hypertextual links to the items' respective product information pages.

For multiple-term queries, the query server 132 effectively logically ANDs the query terms together to perform the search. For example, if the user enters the terms "JAVA" and "PROGRAMMING" into the title field 220, the query server 132 will search for and return a list of all items that have both of these terms within the title. Thus, if any query term does not produce a match (referred to herein as a "non-matching term"), the query will produce a NULL query result. Presenting a NULL query result to the user can cause significant user frustration. To reduce this problem, in this event, the user may be presented with a list of items that are deemed to be "close matches." Although the search engine described herein logically ANDs the query terms together, it will be recognized that the invention can be applied to search engines that use other methods for processing queries.

In accordance with the invention, the search engine uses the query term correlation data stored in the correlation table 137 to select the related terms that best match the user's query. The search engine then presents the related terms to the user, allowing the user to refine the search and enhance discovery of relevant information. The query term correlation data indicates relationships between query terms, and is used to effectively predict query terms that are likely to be helpful to the search refinement process. In accordance with another aspect of the invention, the correlation table 137 preferably contains or reflects historical information about the frequencies with which specific query terms have appeared together within the same query.

The general format of the correlation table 137 is illustrated in FIG. 1. In the embodiment depicted in FIG. 1 and described in detail herein, the correlations between query terms are based solely on frequency of occurrence within the same query. As described below, other types of query term correlations can additionally be used. In addition, although the disclosed implementation uses a table to store the query term correlation data, other types of databases can be used.

As illustrated by FIG. 1, each entry within the correlation table 137 (two entries shown) has two primary components: a "key" term 140, and a "related terms" list 142 for that key term. The related terms list 142 is a list of the N (e.g. 50) query terms that have appeared within the same query as the keyword with the highest degree of frequency, and is ordered according to frequency. For example, the entry for the key term COSMOS (ignoring the single-term prefixes, which are discussed below) is:

COSMOS: ASTRONOMY, SAGAN, UNIVERSE,

indicating that ASTRONOMY has appeared together with COSMOS with the highest degree of frequency; SAGAN has appeared with COSMOS with the second highest degree of frequency, and so on. Each term that appears within the related terms list 142 is deemed to be related to the corresponding key term 140 by virtue of the relatively high frequency with which the terms have occurred within the same query.

As further depicted by FIG. 1, each related term and each key term 140 preferably includes a single-character field prefix which indicates the search field 210, 220, 240 to which the term corresponds. These prefixes may, for example, be as follows: A=author, T=title, S=subject, R=artist, L=label, G=generic. In addition, each related term is stored together with a correlation score 146 which, in the preferred embodiment, indicates the number of times the related term has appeared in combination with the key term

(within the search fields indicated by their respective field prefixes), not counting queries that produced a NULL query result.

Thus, for example, the related term (including prefix) S-ASTRONOMY has a correlation score of **410** under the key term T-COSMOS, indicating that four hundred and ten "successful" queries were received (during the time period to which the table **137** corresponds) which included the combination of COSMOS in the title field and ASTRONOMY in the subject field. Although the field prefixes and correlation scores **146** carry information which is useful to the related terms selection process (as described below), such information need not be preserved.

In operation, when a user submits a query to the web site **130**, the web server **131** passes the query to the query server **132**, and the query server applies the query to the bibliographic database **133**. If the number of items found exceeds a certain threshold (e.g., 50), the query server **132** invokes its related term selection process ("selection process") **139** to attempt to identify one or more related terms to suggest to the user. The selection process may alternatively be invoked without regard to whether a certain item count has been reached.

For each term in the query, the selection process **139** retrieves the respective related terms list **142** (if any) from the correlation table **137**, and if multiple lists result, merges these lists together. The selection process **139** then takes a predetermined number (e.g. 5) of the related terms from the top of the resulting list, and passes these "suggested terms" to the web server **131** with the query result listing. Finally, the web server **131** generates and returns to the user a query result page (FIG. **9**) which presents the suggested terms to the user for selection.

In one embodiment, the related terms lists are merged by retaining only the intersecting terms (terms which are common to all lists), and discarding all other terms. An important benefit of this method is that any single related term of the resulting list can be added to the query without producing a NULL query result. To take advantage of this feature, these related terms are preferably presented to the user using an interface method (as in FIG. **9**) which requires the user to add only one related term to the query per query submission.

The operation of the related term selection process **139** is described in further detail below.

The disclosed search engine also preferably uses historical query submissions and item selections to rank query results for presentation to the user. A preferred method for ranking query results based on such data is disclosed in U.S. patent application Ser. No. 09/041,081 filed Mar. 10, 1998. The search engine also preferably uses correlations between query terms to correct misspelled terms within search queries. A preferred method for correcting spelling errors in search queries is disclosed in U.S. patent application Ser. No. 09/115,662 entitled "System and Method for Correcting Spelling Errors in Search Queries," filed Jul. 15, 1998. The disclosures of these applications are hereby incorporated by reference.

II. Capturing and Processing of Query Information

As indicated above, the query term correlation data is preferably generated from the query log **135** using the table generation process ("generation process") **136**. In the preferred embodiment, the table generation process **136** is implemented as an off-line process which runs once a day and generates a new query correlation table **137**. The process effectively generates the table from the M most recent daily query log files **135**(1)–**135**(M). Using a relatively small M (e.g., 5) tends to produce query term correlation data that heavily reflects short term buying trends (e.g., new releases, weekly best-sellers, etc.), while using a larger M (e.g., 100) tends to produce a more comprehensive database. A hybrid approach can alternatively be used in which the table is generated from a large number of log files, but in which the most recent log files are given greater weight. For example, queries submitted during the last week can be counted three times when generating the correlation scores **146**, while queries submitted from one week to one month ago can be counted only once. In addition, rather than using M consecutive days of query submissions, the generation process **136** could use samples of query submissions from multiple different time periods.

In the preferred embodiment, the building of the query correlation table **137** consists of two primary phases: (1) generating daily log files, and (2) periodically parsing and processing these log files to generate the query correlation table **137**. Rather than generate new query term correlation data each time log information becomes available, the generation process **136** preferably generates and maintains separate query term correlation data for different constituent time periods of a relatively short length. In the preferred embodiment, the constituent time period is one day such that query term correlation data for a single day is stored in a daily results file. Each time query term correlation data is generated for a new constituent time period, the generation process **136** preferably combines this new data with existing data from earlier constituent time periods to form a collective query correlation table with information covering a longer composite period of time. This process is depicted in FIG. **6** and is described further below.

Any of a variety of alternative methods could be used to generate the correlation table **137**. For example, the generation process **136** could alternatively be implemented to update the query correlation table in real time by augmenting the table each time a user submits a successful query. In addition, the table generation process **136** and/or the selection process **139** could take into consideration other types of correlations between query terms, including extrinsic or "static" correlations that are not dependent upon the actions of users.

A. Generating Daily Query Log Files

A web server generally maintains a log file detailing all of the requests it has received from web browsers. The log file is generally organized chronologically and is made up of several entries, each containing information about a different request.

In accordance with the invention, each time a user performs a search, the web server **131** stores information about the submitted query in a log entry of a query log **135**. In addition, the web server **131** generates daily query log files **135**(1)–**135**(M) which each contain the log entries for a respective day. FIG. **3** illustrates four log entries of a sample daily query log file **135**. Each entry in the log file **135** includes information about a particular HTTP (Hypertext Transfer Protocol) transaction. The first log entry **310** contains date and time information for when the user submitted the query, the user identifier corresponding to the identity of the user (and, in some embodiments, identification of the particular interaction with the web server), the name of the web page where the query was entered, query terms entered by the user, and the number of the items found for the query. The "items_found" values in the log preferably indicate the number items that exactly matched the query.

For example, entry **310** indicates that at 2:23 AM on Feb. 13, 1998, user 29384719287 submitted the query {title=

Snow Crash} from the book search page and that two items were found that exactly matched the query. Entry **320** indicates that the same user selected an item having an ISBN of 0553562614 about twenty seconds later, and that this selection was made from a search results page (as is evident from the HTTP_REFERRER line). Other types of user actions, such as a request to place an item in a shopping cart or to purchase an item, are similarly reflected within the query log **135**. As indicated by the above example, a given user's navigation path can be determined by comparing entries within the query log **135**.

B. Generating the Correlation Table

FIG. **4** shows the preferred method for generating the correlation table **137**. In step **410** the generation process **136** goes through the most recent daily query log file to identify all multiple-term queries (i.e., queries comprised of more than one term) that returned at least one item ("items_found">0) in the query result. In step **420**, the generation process **136** correlates each query ("key") term found in the set of queries to related terms that were used with the key term in a particular query, and assigns the related term a correlation score **146**. The correlation score indicates the frequency with which specific terms have historically appeared together within the same query during the period reflected by the daily query log. In step **430**, the generation process **136** stores the terms coupled with their correlation scores in a daily results file. In step **440**, the generation process **136** merges the daily results files for the last M days. Finally, in step **450**, the generation process **136** creates a new correlation table **137** and replaces the existing query correlation table.

In the preferred embodiment, the generation process **136** is executed once per day at midnight, just after the most recent daily query log is closed. In addition, it is assumed that the M-1 most recent daily query logs have already been processed by steps **410–430** of the process to generate respective daily results files.

Each of the steps **410–450** of the FIG. 4 process will now be described in greater detail.

Step 1: Processing the Daily Query Log File

As indicated above, the generation process **136** parses the daily query log file in step **410** to identify and extract successful multi-term queries. Ignoring the query submissions that produced a NULL query result (items_found=0) provides the important benefits of (1) preventing non-matching terms from being added to the correlation table—either as keywords or as related terms—and (2) excluding potentially "weak" correlations between matching terms from consideration. In addition, as described below, excluding such "unsuccessful" query submissions enables the query terms selection process **139** to be implemented so as to guarantee that the modified query will produce a successful query result (i.e., a query result in which the item count is greater than zero).

Using the FIG. **3** log sequence as an example, the generation process **136** would parse the sample daily query log file **135** beginning with log entry **310**. The generation process **136** would extract the query for the first log entry **310** because the query contains more than one query term and "items_found" is greater than zero. Next, the generation process **136** would ignore entry **320** because it contains no query terms. The generation process **136** would then ignore entry **330** because although there are multiple query terms, the number of items found is not greater than zero. The generation process **136** would next extract the log entry **340** and continue through the daily query log file **135**. In some embodiments, other information such as query field or subsequent actions performed by the user may be used to determine which query submissions to extract or how heavily the queries should be weighted. In addition, other methods may be used to extract the information from the query log.

Step 2: Correlate Terms

In accordance with the invention, the generation process **136** first takes each extracted query, and for each query term, adds a single-character field prefix ("prefix") which indicates the search field in which the query term was entered. Thus, for example, using the prefixes listed above, the prefix "T" would be added to the terms "SNOW" and "CRASH," in log entry **310**, and the prefix "S" would be added to the terms "OUTDOOR" and "TRAIL," in log entry **340**. During this process, identical terms that were submitted in is different search fields are assigned different prefixes and are treated as different terms. For example, the term "SNOW" with a prefix of "T" would be treated as different from "SNOW" with the prefix "S." In the implementation described herein, the key term and related terms are stored without regard to alphabetic case, although case information can alternatively be preserved.

The generation process **136** then maps each query ("key") term found in the query and its prefix to other terms ("related terms") used with that particular query. A correlation score is maintained for each related term in the mapping based on the number of times the related term occurred in combination with the key term. The final values of the correlation scores taken over M days are stored within the query correlation table **137** as the correlation scores **146** depicted in FIG. **1**.

For example, if a user submits the query "ROUGH GUIDE TO LONDON," in the title field **220**, the terms would first be coupled with the prefix "T." The correlation scores in the mapping for "T-GUIDE," "T-TO," and "T-LONDON," relative to the key "T-ROUGH," would be incremented. Similarly, the correlation scores for the related terms under the keys "T-GUIDE," "T-TO," and "T-LONDON" would also be incremented.

FIG. **5A** illustrates an example mapping. In this figure, it is assumed that the generation process **136** has already processed many thousands of log entries. For each key term **140** stored in the table **137A**, there is a related terms list **142** such that each related term in the list is coupled with a prefix and a value **146** representing the correlation score. Each time the key term **140** and a related term **142** are used together in a query, the related term's value **146** is incremented.

Assume that the table generation process **136** parses a query "OUTDOOR BIKE TRAIL" submitted in the subject field. FIG. **5A** shows the mapping before the query is added. In response to the query, the generation process **136** updates the mapping **137A** producing the mapping **137B** shown in FIG. **5B**. The generation process **136** first looks up the key term "S-OUTDOOR" **560** and then looks for the related terms "S-BIKE" **580** and "S-TRAIL" **590**. If the related term is found, its value is incremented. If the related term is not found, the generation process **136** adds the related term and assigns it a beginning value. In the example shown in FIG. **5B**, the values for both "S-BIKE" **580** and "S-TRAIL" **590** have been incremented by one. Note that under the key term "T-OUTDOOR," the value for the term "S-TRAIL" was incremented while the value for the term "T-TRAIL" was not incremented. This is because the query was submitted in the subject field, thus affecting only terms with the prefix "S."

In some embodiments, certain key terms may be excluded from the mapping if they are frequently used, and yet do not further the search refinement process. For example, common articles such as "THE," "A," "TO," and "OF" may be excluded from the mapping. While only three partial entries are depicted in FIG. **5A**, many thousands of entries would be stored in a typical daily results file. In the preferred implementation, the mapping for a daily query log file is stored in a B-tree data structure. In other embodiments, a linked list, database, or other type of data structure can be used in place of the B-tree.

In addition, the amount by which the correlation scores are incremented may be increased or decreased depending on different kinds of selection actions performed by the users on items identified in query results. These may include whether the user displayed additional information about an item, how much time the user spent viewing the additional information about the item, how many hyperlinks the user followed within the additional information about the item, whether the user added the item to his or her shopping basket, and whether the user ultimately purchased the item. For example, a given query submission can be counted twice (such as by incrementing the correlation score by two) if the user subsequently selected an item from the query result page, and counted a third time if the user then purchased the item or added the item to the shopping basket. These and other types of post-search activities reflect the usefulness of the query result, and can be extracted from the query log **135** using well-known tracing methods.

Step 3: Create Daily Results File

Once the mapping is complete, that is, all entries in the daily query log file have been parsed, the generation process **136** creates a daily results file (step **430**) to store the B-tree. In other embodiments, the daily results file may be generated at an earlier stage of the process, and may be incrementally updated as the parsing occurs.

Step 4: Merge Daily Results Files

In step **440**, the generation process **136** generates the query correlation table **137** for a composite period by combining the entries of the daily results files for the length of the composite period. As depicted in FIG. **6**, the table generation process **136** regenerates the query correlation table **137** on a daily basis from the M most recent daily results files, where M is a fixed number such as **10** or **20**. Each day, the daily results file created in step **430** is merged with the last M-1 daily results files to produce the query correlation table **137**

For example, in FIG. **6**, suppose the generation process **136** generates a daily results file for Feb. 7, 1998 **610** and is set to generate a new query correlation table for the period of the last seven days (M=7). At the end of Feb. 7, 1998, the generation process **136** would merge the daily results files from the past seven days for the composite period of Feb. 1, 1998 to Feb. 7, 1998 to form a new query correlation table **137A**. At the end of Feb. 8, 1998, the generation process **136** would generate a daily results file for Feb. 8, 1998 **630** and then merge the daily results files from the past seven days for the composite period of Feb. 2, 1998 to Feb. 8, 1998 to form a new query correlation table **137B**. When the entries are merged, the scores of the corresponding entries are combined, for example, by summing them. In one embodiment, the scores in more recent daily results files are weighted more heavily than those scores in less recent daily results files, so that the query term correlation data more heavily reflects recent query submissions over older query submissions. This "sliding window" approach advantageously produces a query correlation table that is based only on recent query submissions, and which thus reflects the current preferences of users. For example, if a relatively large number of users have searched for the book *Into Thin Air* by Jon Krakauer over the past week, the correlations between the terms "T-INTO," "T-THIN," "T-AIR," and "A-KRAKAUER" will likely be correspondingly high; a query which consists of a subset of these terms will thus tend to produce a related terms lists which includes the other terms.

Step 5: Replace Old Query Correlation Table With New Query Correlation Table

In step **450**, once the daily results files have been merged, the generation process **136** sorts the related terms lists from highest-to-lowest score. The generation process **136** then truncates the related terms lists to a fixed length N (e.g., 50) and stores the query correlation table in a B-tree for efficient lookup. The new query correlation table **137** B-tree is then cached in RAM (random access memory) in place of the existing query correlation table.

III. Using the Table to Generate Related Terms

As indicated above, the query server **132** uses the query correlation table **137** to select related terms to be suggested to the user. More specifically, when a user performs a search which identifies more than a predetermined number of items, the related term selection process ("selection process") **139** returns a query result listing items that match the query along with a set of related terms generated from the query correlation table. An important benefit of this method is that it is highly efficient, allowing the query result page to be returned without adding appreciable delay. Further, the small delay added by the related terms selection process can be completely avoided by optionally generating the related terms concurrently with the search of the bibliographic database **133** (rather than waiting to see if a threshold item count is reached).

FIG. **7** illustrates the sequence of steps performed by the selection process **139**. The selection process **139** first enters a loop (steps **710–740**) in which the selection process **139** looks up a query term in the correlation table and then retrieves the term's related terms list **142**. This continues for each term in the query. Next, if the query has multiple terms, in step **760**, the selection process **139** combines the related terms lists. The lists are preferably combined by taking the intersection of the related terms lists (i.e., deleting terms which do not appear in all lists) and summing the correlation scores of the remaining terms. At this point, every term which remains in the list is a term which has appeared, in at least one prior, successful query, in combination with every term of the present query. Thus, assuming entries have not been deleted from the bibliographic database **133** since the beginning of the composite time period (the period to which the table **137** applies), any of these terms can be added individually to the present query without producing a NULL query result. In other embodiments, the selection process **139** combines the related terms lists by summing the correlation scores of terms common to other related terms lists, without deleting any terms. Another implementation might give weighted scores for intersecting terms such that terms appearing in more than one related terms list are weighted heavier than those terms appearing only in a single related terms list.

In step **770**, the selection process **139** selects the X terms with the highest values from the list, where X can be any desired number. In one embodiment, the selection process **139** chooses the top X related terms without regard to the field prefixes of these related terms. The selection process may alternatively be configured to select only those related terms that correspond to the search field(s) of the present query; for example, if the query was entered into the subject field **240** (FIG. **2**), the user may be presented only with other subject terms (related terms with the prefix "S").

For single-term queries, the selection process **139** thus retrieves the top X terms from the table. FIG. **8**A illustrates the related terms that would be generated for a single-term query of "TRAIL" in the subject field using the mapping from FIG. **5**B. The selection process **139** would look up the key term "S-TRAIL" **570** and select X related terms with the highest X values. For example, suppose the selection process **139** were configured to suggest three related terms (X=3) that correspond to the search field(s) of the present query. The selection process **139** would then look up the key term "S-TRAIL" **570** and display the three related terms with the top three values **810** and with the same prefix as the key term, as illustrated in FIG. **8**A.

For multiple-term queries, the selection process **139** obtains the related terms lists **142** for each of the query terms, and then takes the intersection of these lists. FIG. **8**B illustrates the related term results for a multiple-term query in the subject field of "OUTDOOR TRAIL" using the mapping from FIG. **5**B. The selection process **139** would look up the key terms "S-OUTDOOR" **560** and "S-TRAIL" **570** and see if they have any related terms in common. In the mapping, the related terms "S-BIKE," "S-SPORTS," and "S-VACATION" are found under the key terms "S-OUTDOOR" **560** and "S-TRAIL," **570**; thus "S-BIKE," "S-SPORTS," and "S-VACATION" are the intersecting terms **820** as illustrated in FIG. **8**B. The selection process **139** would then display the X intersecting terms with the same prefix and the X highest summed correlation scores. If there were less than X intersecting, related terms, the selection process **139** could show the intersecting terms with any prefix or use other criteria to generate the remaining related terms. For example, the process **139** could take the top Y terms with the highest summed correlation scores from the non-intersecting related terms, although suggesting such terms could produce a NULL query result.

As indicated above, the method can alternatively be implemented without preserving or taking into account search field information. In addition, the method can be appropriately combined with other techniques for generating related terms, including techniques which use the contents of the query result.

IV. Presenting the Related Query Terms to the User

There are a number of different ways to present the related terms to the user, including the conventional methods (check boxes and drop-down menus) described above. In implementations which suggest only the intersecting related terms, an interface which requires the user to add no more than one related term per query submission is preferably used, so that the modified query will not produce a NULL query result.

In the preferred embodiment, the related terms are presented though hypertextual links which combine both the original query term(s) and a respective related term. For example, if the user enters the query "ROUGH" in the subject field, three additional hyperlink may be displayed on the query result page, each of which generates a modified search when clicked on by the user. Each of these links is formed by combining the user's query with a related term (e.g., the three hyperlinks might be "ROUGH—GUIDE," "ROUGH—LONDON," and "ROUGH—TERRAIN"). When the user clicks on one of these links, the corresponding modified query is submitted to the search engine. The method thus enables the user to select and submit the modified query with a single action (e.g., one click of a mouse). As an inherent benefit of the above-described method of generating the related terms, each such link produces as least one "hit."

FIG. **9** illustrates a sample query result page 900 in which a user has performed a subject field search on the terms "OUTDOOR TRAIL" and has received a set of three related terms, each of which is incorporated into a respective hyperlink **910**. The page will also typically contain a listing of the query result items **920**. If the user clicks on the hyperlink "OUTDOOR TRAIL—BIKE," the search engine will perform a search using the terms "S-OUTDOOR," "S-TRAIL," and "S-BIKE," and will then return the associated items. The query result page 900 may also have search fields (not shown) for allowing the user to edit the query.

Any of a variety of additional techniques can be used in combination with this hyperlink-based interface. For example, in one embodiment, the query server **132** automatically selects the related term at the top of related terms list (such as the term "bike" in the FIG. **9** example), and searches the query result to identify a subset of query result items that include this related term. The query server **132** thereby effectively applies the "top" suggested modified query to the bibliographic database **133**. This process could be repeated using additional related terms in the list. The items within the subset can then be displayed to the user at the top of the query result list, and/or can be displayed in highlighted form. Further, the query server **132** could cache the list of items that fall within the subset, so that if the user submits the modified query (such as by clicking on the link "OUTDOOR BIKE—TRAIL" in FIG. **9**), the query server could return the result of the modified search without having to search the bibliographic database. Special tags or codes could be embedded within the modified-query hyperlinks and passed to the web site **130** to enable the query server **132** to match the modified queries to the cached results.

Although this invention has been described in terms of certain preferred embodiments, other embodiments that are apparent to those of ordinary skill in the art are also within the scope of this invention. Accordingly, the scope of the present invention is defined only by reference to the appended claims.

In the claims which follow, reference characters used to denote process steps are provided for convenience of description only, and not to imply a particular order for performing the steps.

What is claimed is:

1. In a computer system that implements a search engine which is accessible to a community of users, a method of assisting users in refining search queries to enhance discovery, the method comprising the computer-implemented steps of:

(a) processing search queries submitted to the search engine by a plurality of users over a period of time to generate query term correlation data, the query term correlation data reflecting frequencies with which query terms appear together within the same search query;

(b) receiving a search query from a user, the search query including at least one query term;

(c) using at least the query term correlation data to identify a plurality of additional query terms that are deemed to be related to the at least one query term; and

(d) presenting the plurality of additional query terms to the user for selection to allow the user to refine the search query.

2. The method of claim **1**, wherein step (a) comprises generating a data structure which that links key terms to related terms based on correlations between occurrences of terms within historical query submissions, and step (c) comprises accessing the data structure to look up related terms.

3. The method of claim **1**, wherein the search query includes multiple query terms, and step (c) comprises the sub-steps of:

(c1) for each of the multiple query terms, identifying a set of terms that have previously occurred in combination with the respective query term within a successful query; and

(c2) selecting, as the additional terms, a set of terms that are common to all of the sets identified in step (c1).

4. The method of claim **3**, wherein step (d) comprises presenting the additional terms via a user interface which inhibits the user from selecting more than one additional term, the method thereby guaranteeing that a modified query produced by adding an additional term will not produce a NULL query result.

5. The method of claim **4**, wherein step (d) comprises presenting the user with a plurality of hyperlinks which can be selected to submit a modified query, each hyperlink adding a different respective additional term to the query.

6. The method of claim **1**, wherein step (a) comprises processing a log that includes search queries submitted to the search engine.

7. The method of claim **6**, wherein the step of processing the log comprises ignoring search queries that produced a NULL query result.

8. The method of claim **6**, wherein the step of processing the log comprises applying a time-based biasing function to the log to favor recent search query submissions over aged search query submissions, so that the query term correlation data and the additional terms reflects current preferences of the community of users.

9. The method of claim **1**, wherein step (a) comprises updating the query term correlation data substantially in real time as the search queries are received by the search engine.

10. The method of claim **1**, wherein step (d) comprises presenting the user with a plurality of hyperlinks, each hyperlink being selectable to submit a refined search query which includes a respective additional query term, the method thereby enabling the user to initiate a refined search with a single action.

11. The method of claim **1**, wherein step (a) further comprises evaluating postquery-submission actions of users to identify search queries that are deemed to have produced useful results, and weighting the search queries that produced useful results more heavily in generating the correlation data.

12. The method of claim **1**, wherein step (c) is performed in parallel with a step of applying the query to a database to be searched.

13. The method of claim **1**, further comprising using at least one of the additional terms to select query result items to display at the top of a query result listing.

14. In a computer system that implements a search engine in which related terms are suggested to users to facilitate interactive refinement of search queries, a system for generating related terms, comprising:

a first process which generates a data structure that links key terms to related terms based at least upon correlations between occurrences of terms within historical query submissions; and

a second process which uses the data structure in combination with a search query submitted by a user to select related terms to suggest to the user.

15. The system of claim **14**, wherein the first process determines the correlations between occurrences of terms by at least parsing a log that includes historical query submissions.

16. The system of claim **14**, wherein the first process generates and updates the data structure substantially in real-time as search queries are received by the search engine.

17. The system of claim **14**, wherein the first process regenerates the data structure periodically from a log of recent query submissions, so that the related terms suggested to the user reflect current preferences of users.

18. The system of claim **14**, wherein the first process determines the correlations by at least counting a number of times the terms have occurred within the same query.

19. The system of claim **14**, wherein the first process ignores query submissions that produced NULL query results, so that the data structure reflects only successful query submissions.

20. The system of claim **19**, wherein the second process processes a multiple-term search query by at least:

(a) for each term in the search query, using the data structure to identify a respective set of terms that were previously submitted to the search engine in combination with the term in a successful search query; and

(b) selecting a set of related terms such that each related term is common to each set identified in step (a).

21. The system of claim **20**, further comprising a user interface process which presents the set of related terms to the user for selection such that no more than one related term can be added to the search query per query submission, the second process thereby ensuring that a modified query produced by adding a related term will not produce a NULL query result.

22. In a computer system that implements a search engine that is accessible to a community of users, a method of assisting users in refining search queries to enhance discovery, the method comprising:

(a) receiving a search query from a user, the search query including at least one query term;

(b) using at least historical search query data to identify a plurality of additional query terms that are deemed to be related to the at least one query term, the historical search query data based on previously submitted search queries; and

(c) presenting the plurality of additional query terms to the user for selection to allow the user to refine the search query.

23. The method of claim **22**, wherein the search query includes multiple query terms, and step (b) comprises the sub-steps of:

(b1) for each of the multiple query terms, identifying a set of terms that have previously occurred in combination with the respective query term within a successful query; and

(b2) selecting, as the additional query terms, a set of terms that are common to all of the sets identified in step (b1).

24. The method of claim **23**, wherein step (d) comprises using a user interface method which inhibits the user from

selecting more than one additional term, the method thereby guaranteeing that a modified query produced by adding an additional term will not produce a NULL query result.

25. In a search engine that suggests related terms to users to facilitate search refinement, a method of generating related terms so as to increase a likelihood that a modified query will not produce a NULL query result, the method comprising:

(a) receiving a search query from a user, the query including at least one term;

(b) for each term in the search query, using historical query information to identify a respective set of terms that were previously submitted to the search engine, in combination with the term, in a successful search query;

(c) selecting a set of related terms such that each related term is common to each set identified in step (b); and

(d) presenting the set of related terms to the user for addition to the search query.

26. The method of claim 25, wherein step (d) comprises presenting the related terms via a user interface which inhibits the user from selecting more than one additional term to add to the query.

27. The method of claim 26, wherein the step (d) comprises presenting the user with a plurality of hyperlinks, each hyperlink being selectable to submit a refined search query which includes a respective related term, the method thereby enabling the user to initiate a refined search with a single action.

28. The method of claim 25, wherein the search query comprises multiple query terms.

* * * * *

EXHIBIT 2

US006169986B1

(12) **United States Patent**
Bowman et al.

(10) **Patent No.: US 6,169,986 B1**
(45) **Date of Patent: *Jan. 2, 2001**

(54) **SYSTEM AND METHOD FOR REFINING SEARCH QUERIES**

(75) Inventors: **Dwayne E. Bowman**, Woodinville; **Ruben E. Ortega**; **Michael L. Hamrick**, both of Seattle; **Joel R. Spiegel**, Woodinville; **Timothy R. Kohn**, Seattle, all of WA (US)

(73) Assignee: **Amazon.com, Inc.**, Seattle, WA (US)

(*) Notice: Under 35 U.S.C. 154(b), the term of this patent shall be extended for 0 days.

This patent is subject to a terminal disclaimer.

(21) Appl. No.: **09/411,441**

(22) Filed: **Oct. 1, 1999**

Related U.S. Application Data

(63) Continuation of application No. 09/145,360, filed on Sep. 1, 1998, now Pat. No. 6,006,225.

(60) Provisional application No. 60/089,244, filed on Jun. 15, 1998.

(51) **Int. Cl.**[7] **G06F 17/30**
(52) **U.S. Cl.** **707/5**; 707/2; 707/4; 707/10
(58) **Field of Search** 707/5, 2, 10, 4

(56) **References Cited**

U.S. PATENT DOCUMENTS

5,675,819	*	10/1997	Schuetze	704/10
5,721,897	*	2/1998	Rubinstein	707/2
5,787,422	*	7/1998	Tukey et al.	707/5
5,794,233	*	8/1998	Rubinstein	707/4
5,864,845		1/1999	Voorhees et al. .	
5,865,845	*	1/1999	Voorhees et al.	7074/5
5,911,140	*	6/1999	Tukey et al.	707/5
5,913,215	*	6/1999	Rubinstein	707/10
6,006,225	*	12/1999	Bowman et al.	707/5

OTHER PUBLICATIONS

Abstract of *Generating Advanced Query Interfaces*, Lee, Srivastava and Vista, Computer Networks and ISDN System Conference Title: Comput. Netw. ISDN Syst. (Netherlands) vol. 30, No. 1–7, pp. 656–657 (1998).
Abstract of *Using Combination of Evidence for Term Expansion*, Wilkinson, Information Retrieval Research, Proceedings of the 19th Annual BCS–IRSG Colloquium on IR Research (1997).
Abstract of Inquirus, *the NECI Meta Search Engine*, Lawrence and Giles, Computer Networks and ISDN Systems Conference Title: Comput. Netw. ISDN Syst. (Netherlands) vol. 30, No. 1–7, pp. 95–105 (1998).
Abstract of *Facilitating Complex Web Queries Through Visual User Interfaces and Query Relaxation*, Li and Shim, Computer Networks and ISDN Systems Conference Title: Comput. Netw. ISDN Syst. (Netherlands) vol. 30, No. 1–7, pp. 149–159 (1998).

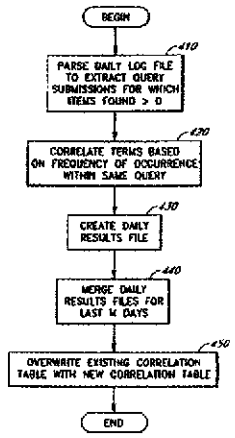
(List continued on next page.)

Primary Examiner—Thomas G. Black
Assistant Examiner—Frantz Coby
(74) *Attorney, Agent, or Firm*—Knobbe, Martens, Olson & Bear, LLP

(57) **ABSTRACT**

A search engine is disclosed which suggests related terms to the user to allow the user to refine a search. The related terms are generated using query term correlation data which reflects the frequencies with which specific terms have previously appeared within the same query. The correlation data is generated and stored in a look-up table using an off-line process which parses a query log file. The table is regenerated periodically from the most recent query submissions (e.g., the last two weeks of query submissions), and thus strongly reflects the current preferences of users. Each related term is presented to the user via a respective hyperlink which can be selected by the user to submit a modified query. In one embodiment, the related terms are added to and selected from the table so as to guarantee that the modified queries will not produce a NULL query result.

12 Claims, 10 Drawing Sheets



OTHER PUBLICATIONS

A User–centred Evaluation of Ranking Algorithms for Interactive Query Expansion, Efthimiadis, Proceedings of the 16th Annual International ACM SIGIR Conference, Pittsburgh, pp. 146–159 (1993).

Concept Based Query Expansion, Qiu and Frei, Proceedings of the 16th Annual International ACM SIGIR Conference, Pittsburgh, pp. 160–169 (1993).

Improving Retrieval Performance by Relevance Feedback, Salton and Buckley, J. of Am. Society for Info. Science 41(4):288–297 (1990).

Query Expansion Using Domain–Adapted, *Weighted Thesaurus in an Extended Boolean Model*, Kwon, Kim and Choi, Proceedings of the 3rd International Conference on Information and Knowledge Management (CIKM'94), pp. 140–146 (1994).

Browsing Through Querying: Designing for Electronic Books, Charoenkitkarn, Tam, Chignell and Golovchinsky, at the 5th ACM Conference on Hypertext, Seattle, WA 206–216 (1993).

A Survey of Information Retrieval and Filtering Methods, Faloutsos and Oard, Univ. of Maryland, 22 pages (undated).

A Corpus Analysis Approach for Automatic Query Expansion, Gauch and Wang, Proceedings of the 6th International Conference on Information and Knowledge Management, pp. 278–284 (1997).

Discovering Web Access Patterns and Trends by Applying OLAP and Data Mining Technology on Web Logs, Zaiane, Xin and Han, Proceedings of the IEEE Forum on Research and Technology Advances in Digital Libraries (IEEE ADL'98), pp. 19–29 (1998).

Bartell et al., "Automatic Combination of Multiple Ranked Retrieval Systems", Proceedings of SIGIR'94, Jul. 1994, pp. 173–181, Jul. 1994.

Belkin et al., "The Effect of Multiple Query Representations on Information System Performance" Proceedings of SIGIR'93, Jun. 1993, pp. 339–346, Jun. 1993.

Shaw et al., "Combination of Multiple Searches", Proceedings of TREC–3, Apr. 1995, pp. 105–108, Apr. 1995.

QuarterDeck Web Page, Downloaded Sep. 9, 1996, http://aracnid.qdeck.com/qdeck/products/webcompass, Sep. 1996.

Towell, et al. "Learning Collection Fusion Strategies for Information Retrieval", Proceedings of the 12th Annual Machine Learning Conference, pp. 540–548, Jul. 1995.

Voohees, et al., "Learning Collection Fusion Strategies", Proceedings of SIGIR'95, pp. 172–179, Jul. 1995.

Voorhees, et al., "The Collection Fusion Problem" Proceedings of TREC–3, NIST Special Publication 500–225, pp. 95–104, Apr. 1995.

* cited by examiner

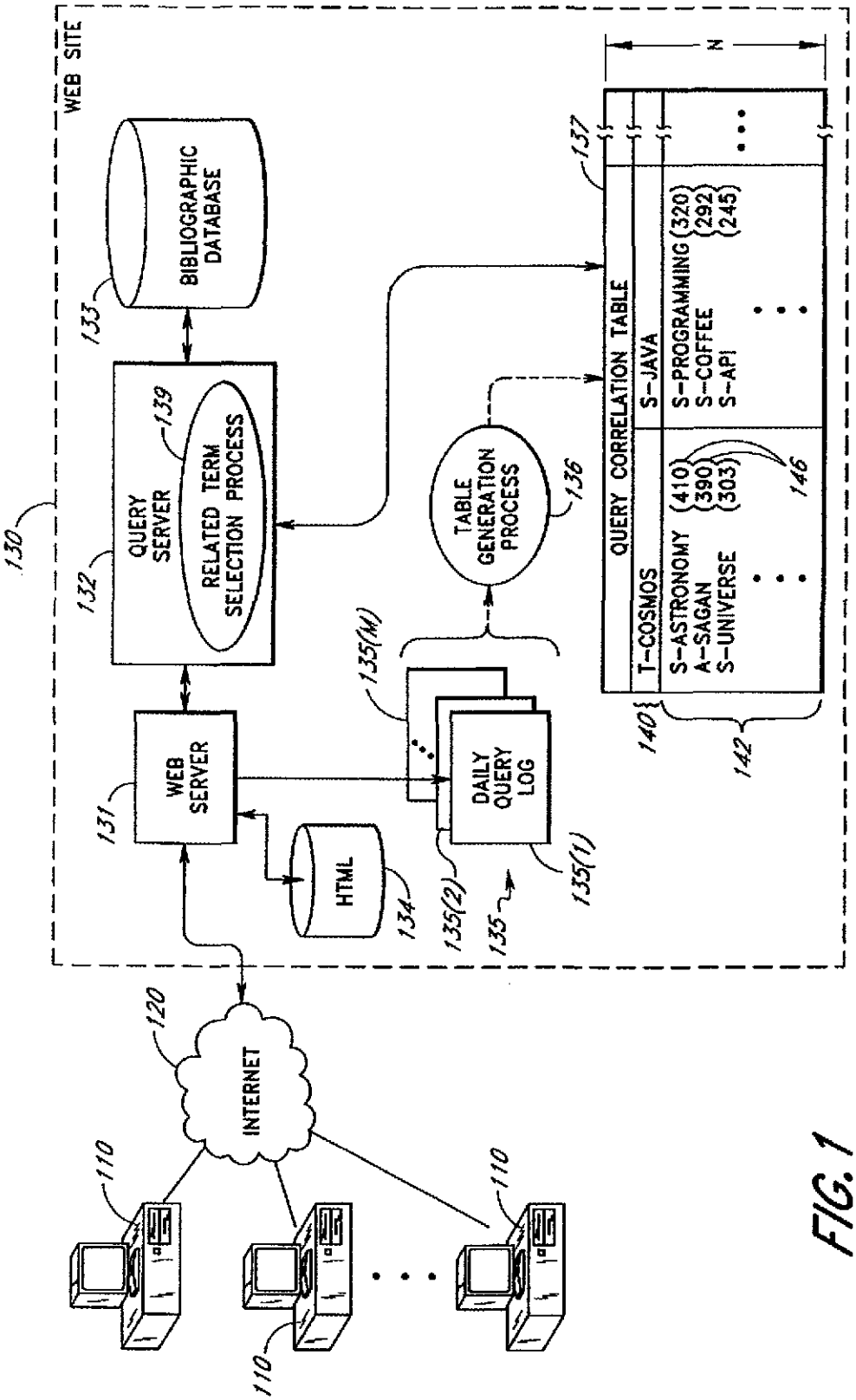

WEB SITE
BIBLIOGRAPHIC DATABASE
133
QUERY SERVER
139
RELATED TERM SELECTION PROCESS
132
130
WEB SERVER
131
HTML
134
DAILY QUERY LOG
135(M)
135(2)
135
135(1)
TABLE GENERATION PROCESS
136
QUERY CORRELATION TABLE
137
N
T-COSMOS
S-JAVA
S-ASTRONOMY (410)
A-SAGAN (390)
S-UNIVERSE (303)
S-PROGRAMMING (320)
S-COFFEE (292)
S-API (245)
146
140
142
INTERNET
120
110
110
110

FIG. 1

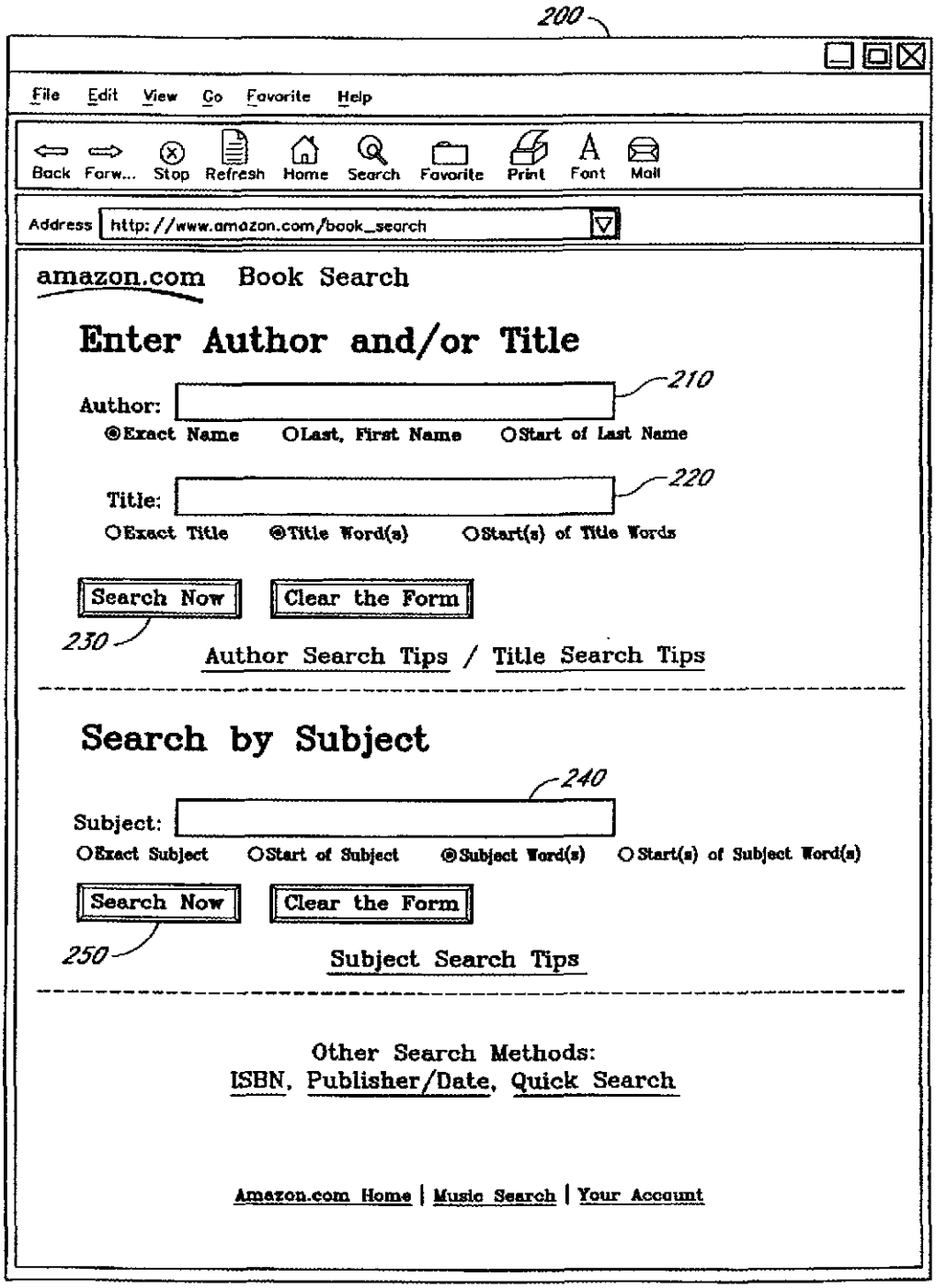

200
File Edit View Go Favorite Help
Back Forw... Stop Refresh Home Search Favorite Print Font Mail
Address http://www.amazon.com/book_search
amazon.com Book Search
Enter Author and/or Title
Author:
210
Exact Name Last, First Name Start of Last Name
Title:
220
Exact Title Title Word(s) Start(s) of Title Words
Search Now
Clear the Form
230
Author Search Tips / Title Search Tips
Search by Subject
240
Subject:
Exact Subject Start of Subject Subject Word(s) Start(s) of Subject Word(s)
Search Now
Clear the Form
250
Subject Search Tips
Other Search Methods:
ISBN, Publisher/Date, Quick Search
Amazon.com Home | Music Search | Your Account

FIG. 2

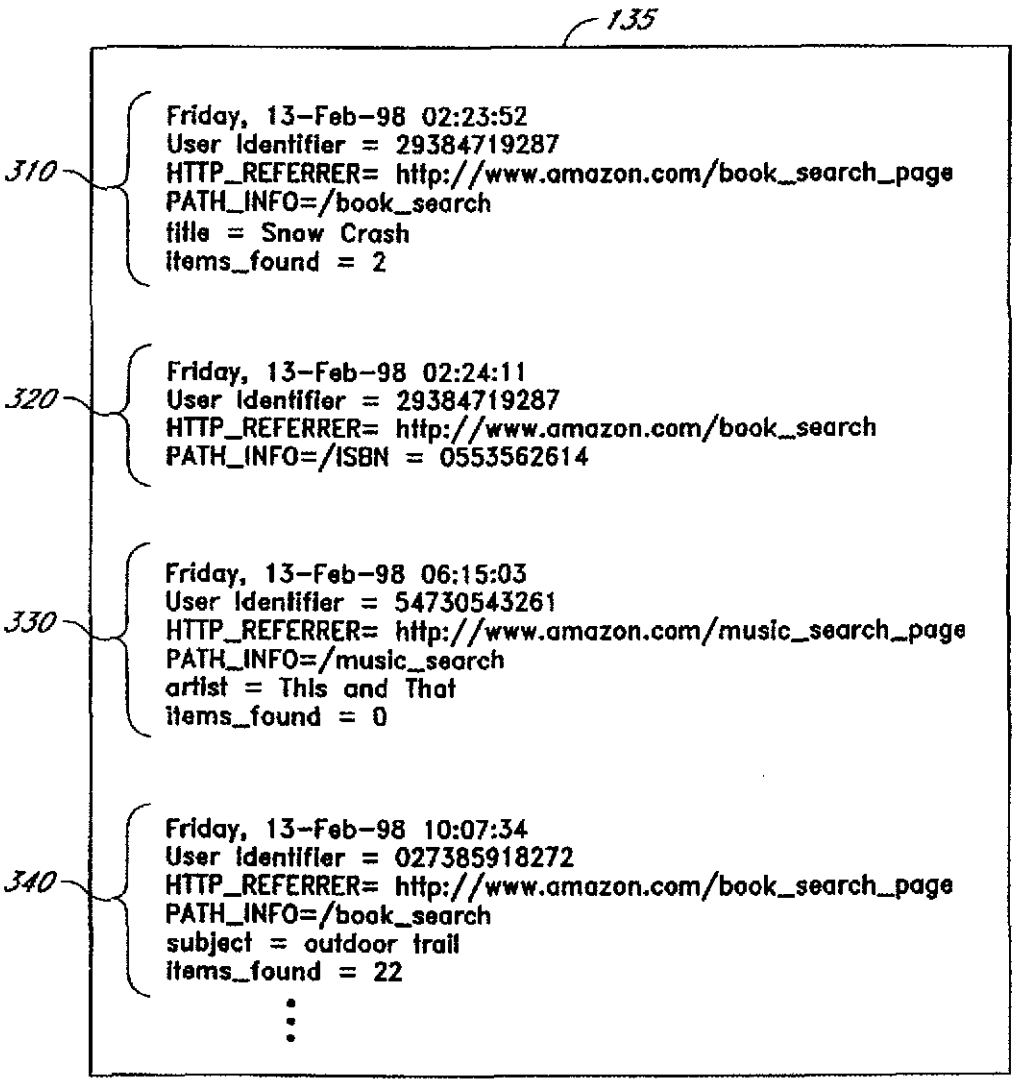

135
310
Friday, 13-Feb-98 02:23:52
User Identifier = 29384719287
HTTP_REFERRER= http://www.amazon.com/book_search_page
PATH_INFO=/book_search
title = Snow Crash
items_found = 2
320
Friday, 13-Feb-98 02:24:11
User Identifier = 29384719287
HTTP_REFERRER= http://www.amazon.com/book_search
PATH_INFO=/ISBN = 0553562614
330
Friday, 13-Feb-98 06:15:03
User Identifier = 54730543261
HTTP_REFERRER= http://www.amazon.com/music_search_page
PATH_INFO=/music_search
artist = This and That
items_found = 0
340
Friday, 13-Feb-98 10:07:34
User Identifier = 027385918272
HTTP_REFERRER= http://www.amazon.com/book_search_page
PATH_INFO=/book_search
subject = outdoor trail
items_found = 22

FIG. 3

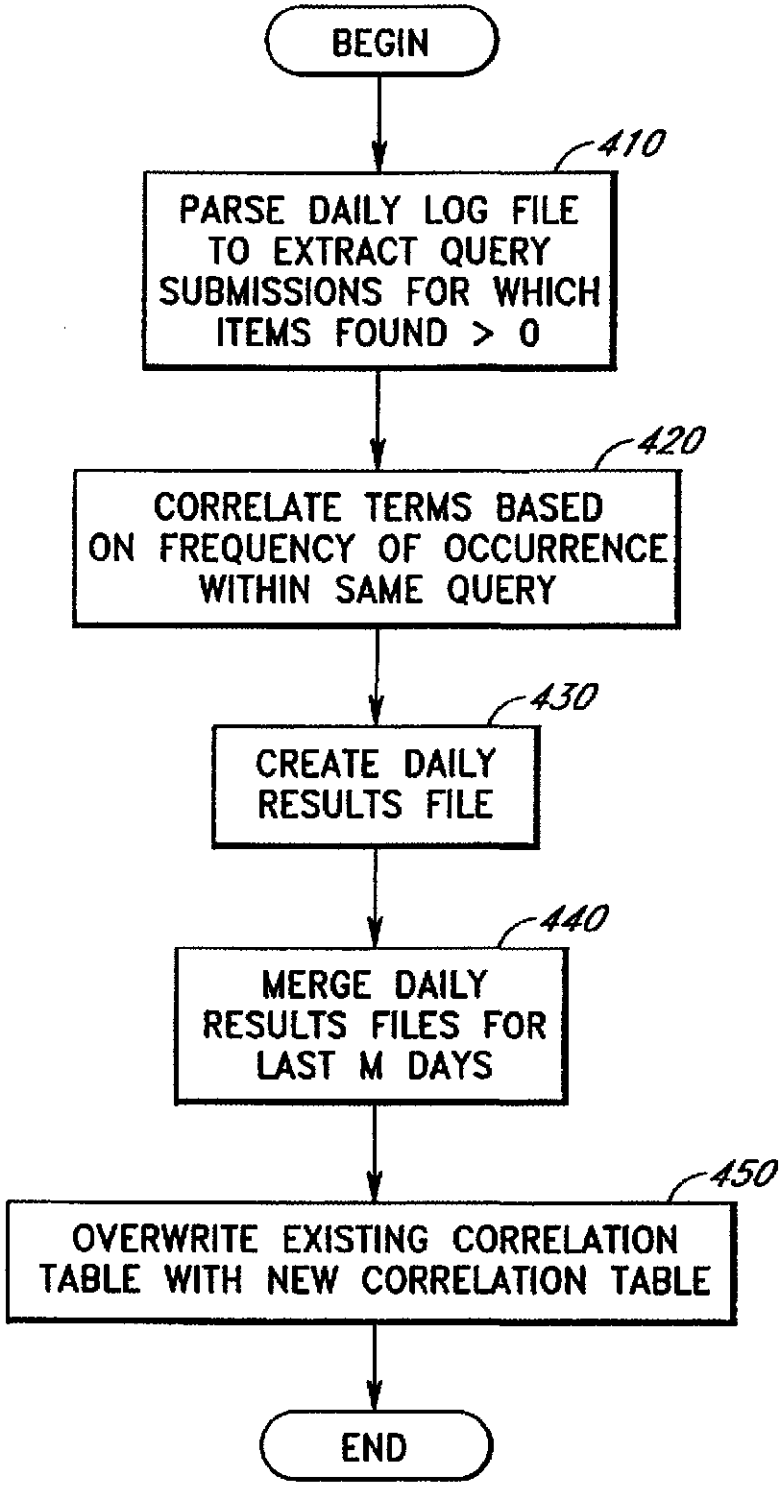

BEGIN
410
PARSE DAILY LOG FILE TO EXTRACT QUERY SUBMISSIONS FOR WHICH ITEMS FOUND > 0
420
CORRELATE TERMS BASED ON FREQUENCY OF OCCURRENCE WITHIN SAME QUERY
430
CREATE DAILY RESULTS FILE
440
MERGE DAILY RESULTS FILES FOR LAST M DAYS
450
OVERWRITE EXISTING CORRELATION TABLE WITH NEW CORRELATION TABLE
END

FIG. 4

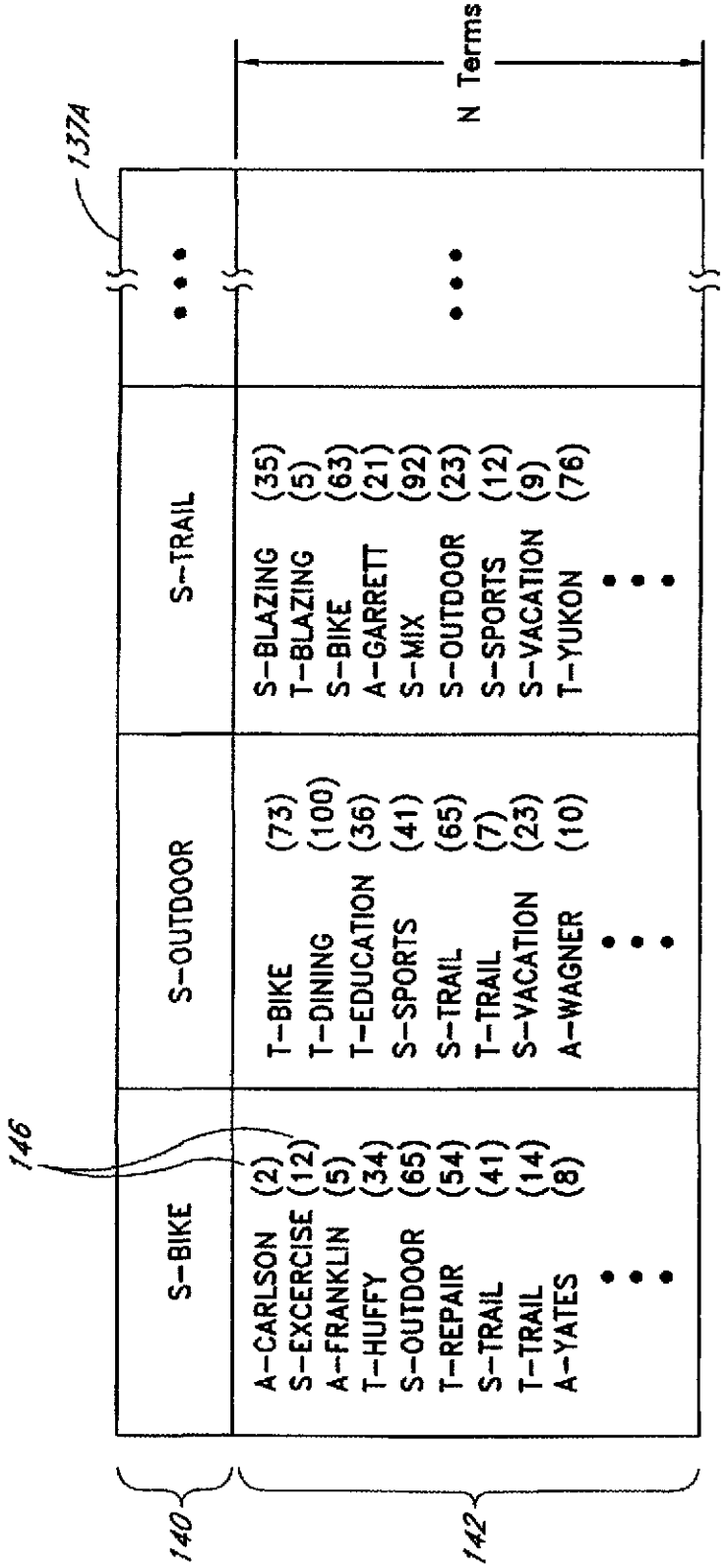

137A
N Terms
S-BIKE
S-OUTDOOR
S-TRAIL
A-CARLSON (2)
S-EXCERCISE (12)
A-FRANKLIN (5)
T-HUFFY (34)
S-OUTDOOR (65)
T-REPAIR (54)
S-TRAIL (41)
T-TRAIL (14)
A-YATES (8)
T-BIKE (73)
T-DINING (100)
T-EDUCATION (36)
S-SPORTS (41)
S-TRAIL (65)
T-TRAIL (7)
S-VACATION (23)
A-WAGNER (10)
S-BLAZING (35)
T-BLAZING (5)
S-BIKE (63)
A-GARRETT (21)
S-MIX (92)
S-OUTDOOR (23)
S-SPORTS (12)
S-VACATION (9)
T-YUKON (76)
146
140
142

FIG. 5A

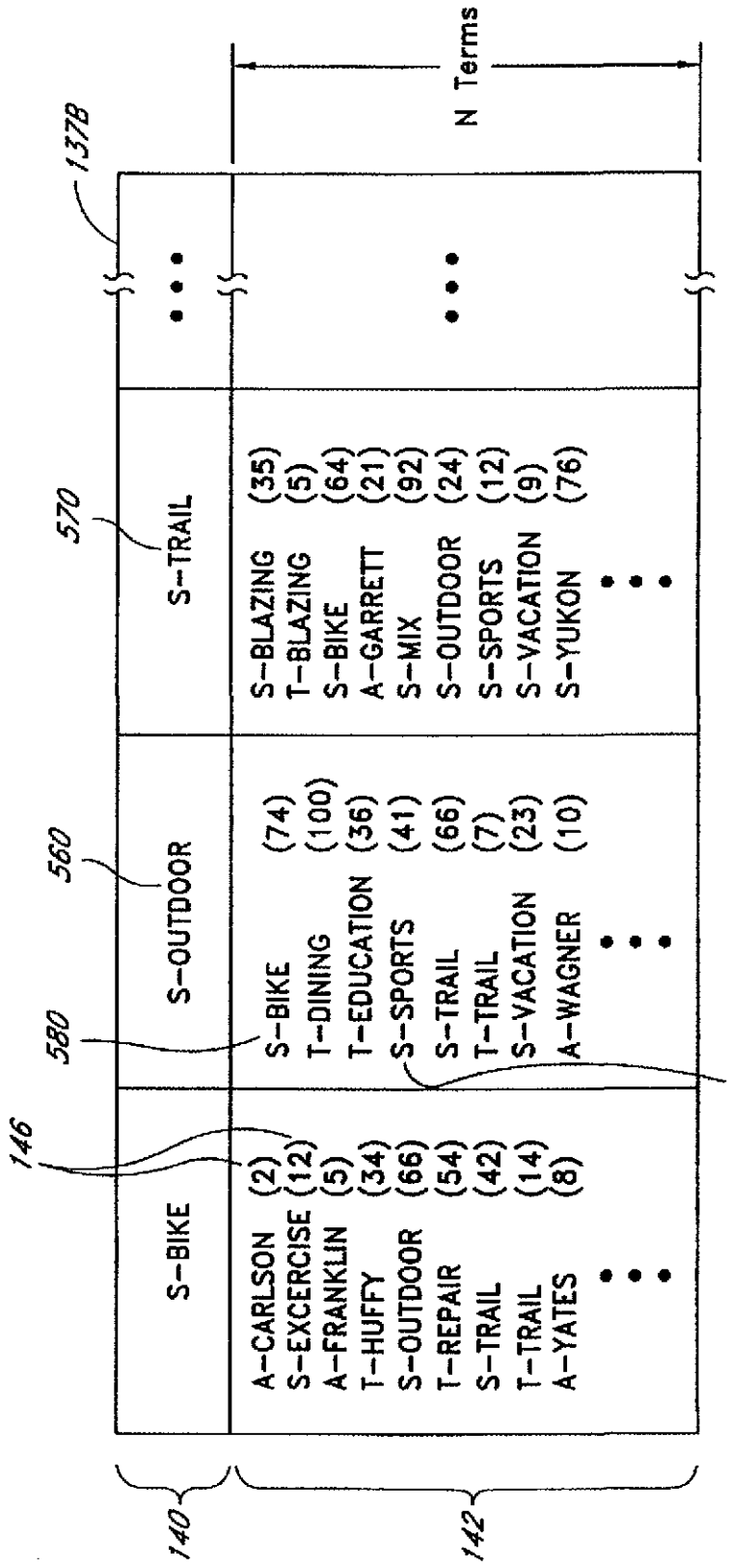

1378
N Terms
570
S-TRAIL
S-BLAZING (35)
T-BLAZING (5)
S-BIKE (64)
A-GARRETT (21)
S-MIX (92)
S-OUTDOOR (24)
S-SPORTS (12)
S-VACATION (9)
S-YUKON (76)
560
S-OUTDOOR
S-BIKE (74)
T-DINING (100)
T-EDUCATION (36)
S-SPORTS (41)
S-TRAIL (66)
T-TRAIL (7)
S-VACATION (23)
A-WAGNER (10)
580
590
146
S-BIKE
A-CARLSON (2)
S-EXCERCISE (12)
A-FRANKLIN (5)
T-HUFFY (34)
S-OUTDOOR (66)
T-REPAIR (54)
S-TRAIL (42)
T-TRAIL (14)
A-YATES (8)
140
142

FIG. 5B

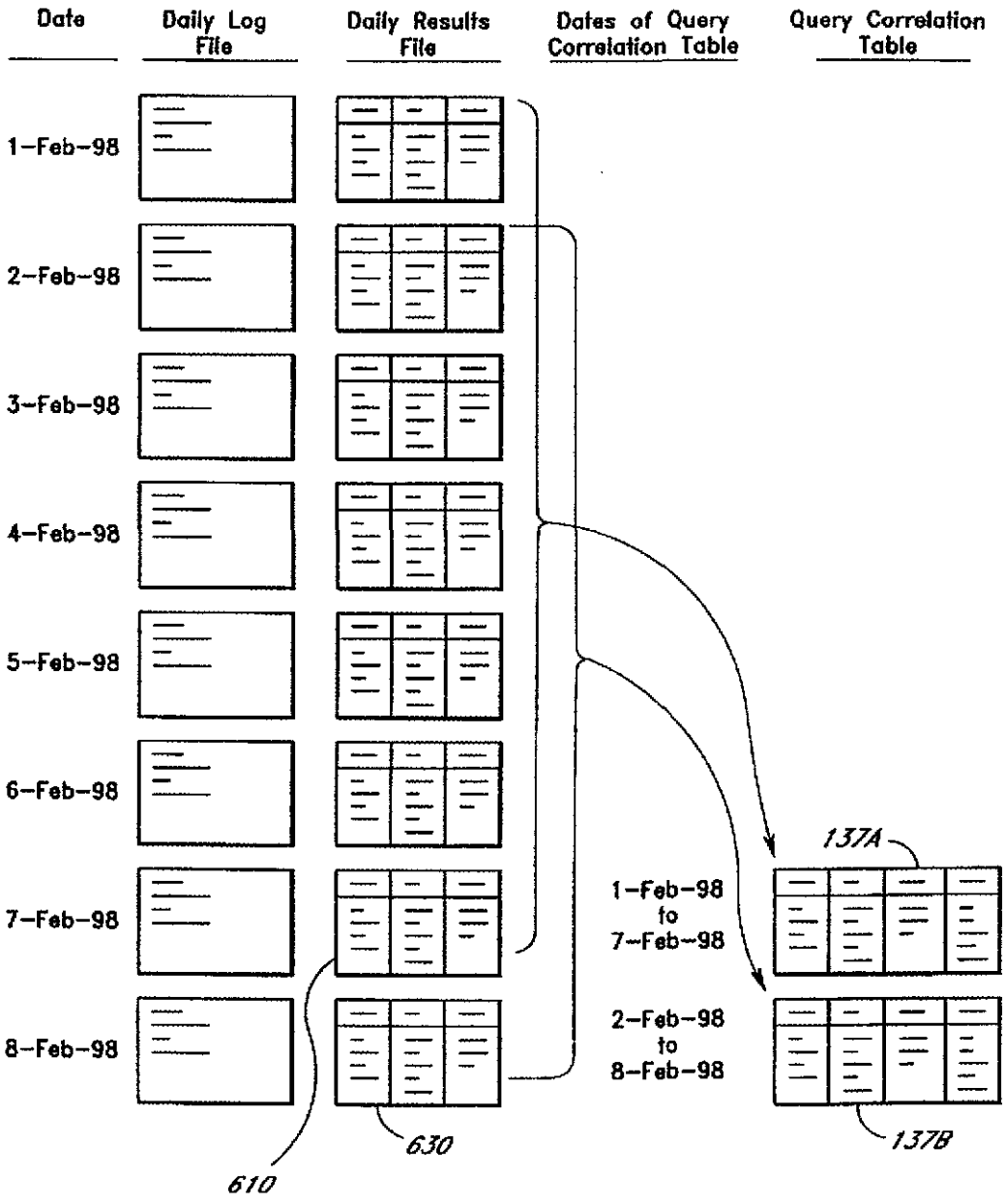

Date
Daily Log File
Daily Results File
Dates of Query Correlation Table
Query Correlation Table
1-Feb-98
2-Feb-98
3-Feb-98
4-Feb-98
5-Feb-98
6-Feb-98
7-Feb-98
8-Feb-98
1-Feb-98 to 7-Feb-98
2-Feb-98 to 8-Feb-98
137A
137B
610
630

FIG. 6

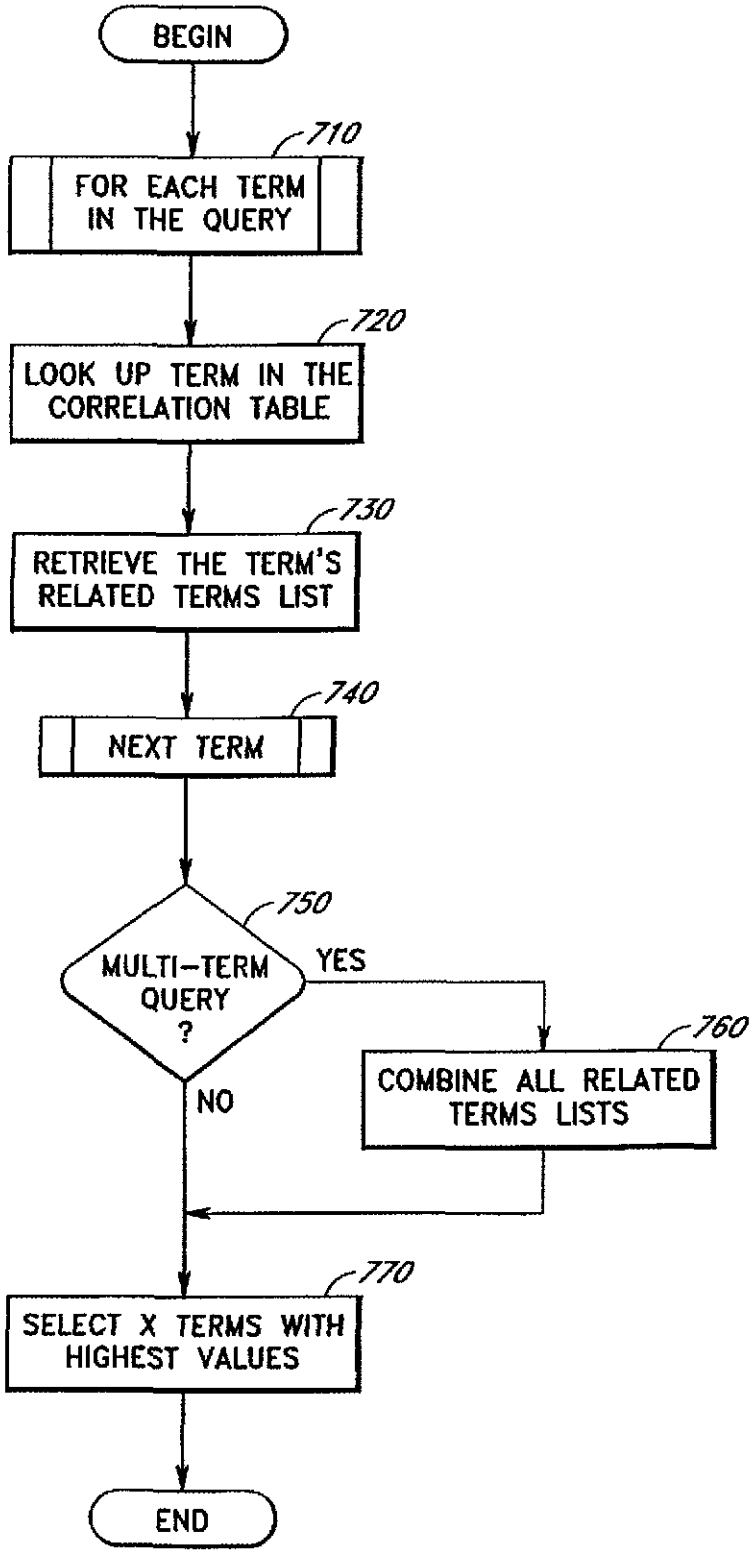

BEGIN
710
FOR EACH TERM IN THE QUERY
720
LOOK UP TERM IN THE CORRELATION TABLE
730
RETRIEVE THE TERM'S RELATED TERMS LIST
740
NEXT TERM
750
MULTI-TERM QUERY ?
YES
NO
760
COMBINE ALL RELATED TERMS LISTS
770
SELECT X TERMS WITH HIGHEST VALUES
END

FIG. 7

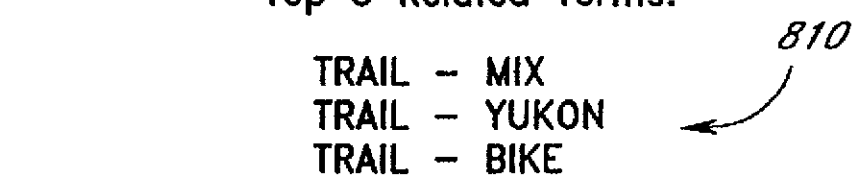
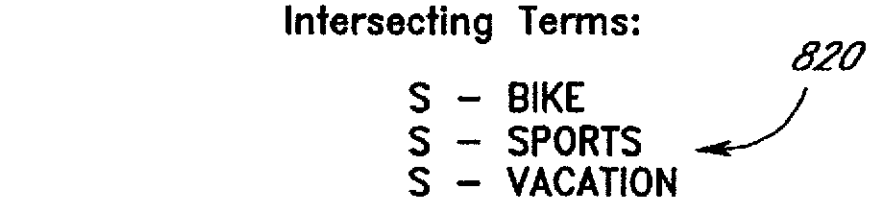

Top 3 Related Terms:
810
TRAIL – MIX
TRAIL – YUKON
TRAIL – BIKE

FIG. 8A


Intersecting Terms:
820
S – BIKE
S – SPORTS
S – VACATION

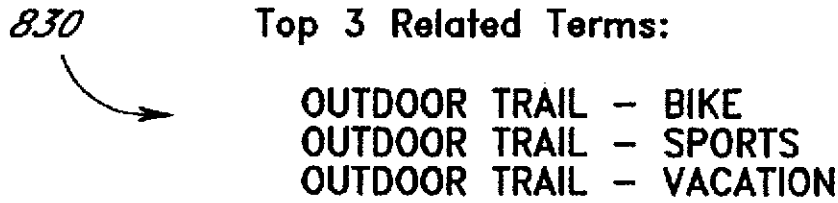

830
Top 3 Related Terms:
OUTDOOR TRAIL – BIKE
OUTDOOR TRAIL – SPORTS
OUTDOOR TRAIL – VACATION

FIG. 8B

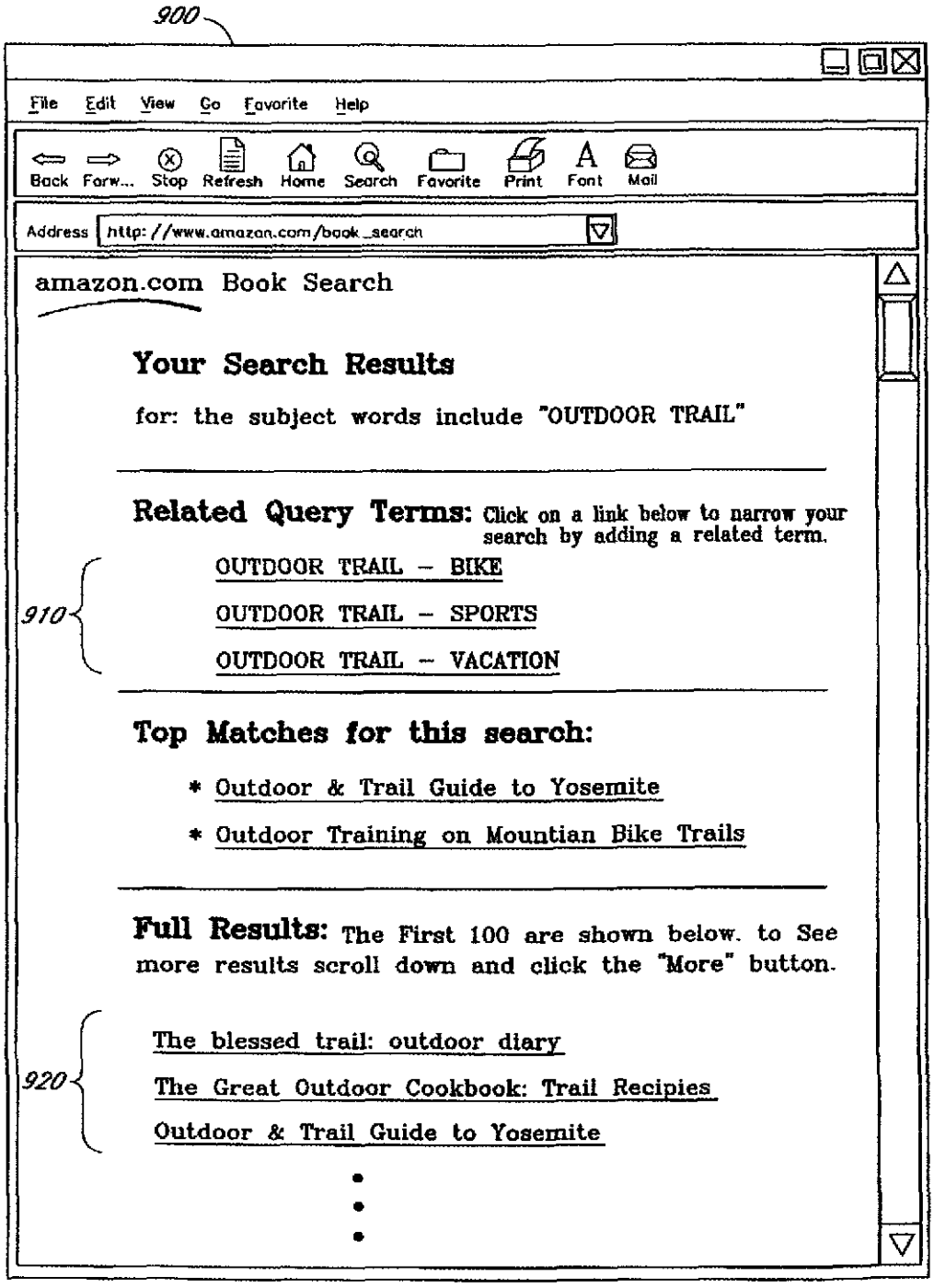

900
File Edit View Go Favorite Help
Back Forw... Stop Refresh Home Search Favorite Print Font Mail
Address http://www.amazon.com/book_search
amazon.com Book Search
Your Search Results
for: the subject words include "OUTDOOR TRAIL"
Related Query Terms: Click on a link below to narrow your search by adding a related term.
910
OUTDOOR TRAIL – BIKE
OUTDOOR TRAIL – SPORTS
OUTDOOR TRAIL – VACATION
Top Matches for this search:
* Outdoor & Trail Guide to Yosemite
* Outdoor Training on Mountian Bike Trails
Full Results: The First 100 are shown below. to See more results scroll down and click the "More" button.
920
The blessed trail: outdoor diary
The Great Outdoor Cookbook: Trail Recipies
Outdoor & Trail Guide to Yosemite

FIG. 9

SYSTEM AND METHOD FOR REFINING SEARCH QUERIES

RELATED APPLICATION

This application is a continuation of application Ser. No. 09/145,360 filed Sep. 1, 1998 now U.S. Pat. No. 6,006,225 claims the benefit of U.S. Provisional Application Ser. No. 60/089,244, filed Jun. 15, 1998, the disclosure of which is hereby incorporated by reference.

BACKGROUND OF THE INVENTION

1. Field of Invention

This present invention relates to query processing, and more specifically relates to techniques for facilitating the process of refining search queries.

2. Description of Related Art

With the increasing popularity of the Internet and the World Wide Web, it is common for on-line users to utilize search engines to search the Internet for desired information. Many web sites permit users to perform searches to identify a small number of relevant items among a much larger domain of items. As an example, several web index sites permit users to search for particular web sites among known web sites. Similarly, many on-line merchants, such as booksellers, permit users to search for particular products among all of the products that can be purchased from the merchant. Other on-line services, such as Lexis™ and Westlaw™, allow users to search for various articles and court opinions.

In order to perform a search, a user submits a query containing one or more query terms. The query may also explicitly or implicitly identify a record field or segment to be searched, such as title, author, or subject classification of the item. For example, a user of an on-line bookstore may submit a query containing terms that the user believes appear within the title of a book. A query server program of the search engine processes the query to identify any items that match the terms of the query. The set of items identified by the query server program is referred to as a "query result." In the on-line bookstore example, the query result is a set of books whose titles contain some or all of the query terms. In the web index site example, the query result is a set of web sites or documents. In web-based implementations, the query result is typically presented to the user as a hypertextual listing of the located items.

If the scope of the search is large, the query result may contain hundreds, thousands or even millions of items. If the user is performing the search in order to find a single item or a small set of items, conventional approaches to ordering the items within the query result often fail to place the sought item or items near the top of the query result list. This requires the user to read through many other items in the query result before reaching the sought item. Certain search engines, such as Excite™ and AltaVista™, suggest related query terms to the user as a part of the "search refinement" process. This allows the user to further refine the query and narrow the query result by selecting one or more related query terms that more accurately reflect the user's intended request. The related query terms are typically generated by the search engine using the contents of the query result, such as by identifying the most frequently used terms within the located documents. For example, if a user were to submit a query on the term "FOOD," the user may receive several thousand items in the query result. The search engine might then trace through the contents of some or all of these items and present the user with related query terms such as "RESTAURANTS," "RECIPIES," and "FDA" to allow the user to refine the query.

The related query terms are commonly presented to the user together with corresponding check boxes that can be selectively marked or checked by the user to add terms to the query. In some implementations, the related query terms are alternatively presented to and selected by the user through drop down menus that are provided on the query result page. In either case, the user can add additional terms to the query and then re-submit the modified query. Using this technique, the user can narrow the query result down to a more manageable set consisting primarily of relevant items.

One problem with existing techniques for generating related query terms is that the related terms are frequently of little or no value to the search refinement process. Another problem is that the addition of one or more related terms to the query sometimes leads to a NULL query result. Another problem is that the process of parsing the query result items to identify frequently used terms consumes significant processor resources, and can appreciably increase the amount of time the user must wait before viewing the query result. These and other deficiencies in existing techniques hinder the user's goal of quickly and efficiently locating the most relevant items, and can lead to user frustration.

SUMMARY OF THE INVENTION

The present invention addresses these and other problems by providing a search refinement system and method for generating and displaying related query terms ("related terms"). In accordance with the invention, the related terms are generating using query term correlation data that is based on historical query submissions to the search engine. The query term correlation data ("correlation data") is preferably based at least upon the frequencies with which specific terms have historically been submitted together within the same query. The incorporation of such historical query information into the process tends to produce related terms that are frequently used by other users in combination with the submitted query terms, and significantly increases the likelihood that these related terms will be helpful to the search refinement process. To further increase the likelihood that the related terms will be helpful, the correlation data is preferably generated only from those historical query submissions that produced a successful query result (at least one match).

In accordance with one aspect of the invention, the correlation data is stored in a correlation data structure (table, database, etc.) which is used to look up related terms in response to query submissions. The data structure is preferably generated using an off-line process which parses a query log file, but could alternatively be generated and updated in real-time as queries are received from users. In one embodiment, the data structure is regenerated periodically (e.g., once per day) from the most recent query submissions (e.g., the last M days of entries in the query log), and thus strongly reflects the current tastes of the community of users; as a result, the related terms suggested by the search engine strongly reflect the current tastes of the community. Thus, for example, in the context of a search engine of an online merchant, the search engine tends to suggest related terms that correspond to the current best-selling products.

In a preferred embodiment, each entry in the data structure is in the form of a key term and a corresponding related terms list. Each related terms list contains the terms which

have historically appeared together (in the same query) with the respective key term with the highest degree of frequency, ignoring unsuccessful query submissions (query submissions which produced a NULL query result). The data structure thus provides an efficient mechanism for looking up the related terms for a given query term.

To generate a set of related terms for refining a submitted query (the "present query"), the related terms list for each term in the present query is initially obtained from the correlation data structure. If this step produces multiple related terms lists (as in the case of a multiple-term query), the related terms lists are preferably combined by taking the intersection between these lists (i.e., deleting the terms that are not common to all lists). The related terms which remain are terms which have previously appeared, in at least one successful query submission, in combination with every term of the present query. Thus, assuming items have not been deleted from the database being searched, any of these related terms can be individually added to the present query while guaranteeing that the modified query will not produce a NULL query result. To take advantage of this feature, the related terms are preferably presented to the user via a user interface that requires the user to add no more than one related term per query submission. In other embodiment, the related terms are selected and displayed without guaranteeing a successful query result.

Because the related terms are identified from previously-generated correlation data, without the need to parse documents or correlate terms, the related terms can be identified and presented to the user with little or no added delay.

One aspect of the invention is thus a method of assisting users in refining search queries. The method is performed by a computer system that implements a search engine that is accessible to a community of users. The method comprises receiving a search query submitted by a user, wherein the search query comprises at least one term. The method further comprises using a history of search queries submitted to the search engine over a selected period of time by the community of users to identify at least one refinement to the search query, and suggesting the at least one refinement to the user.

Another aspect of the invention is a system for assisting users in refining search queries submitted to a search engine. The system comprises a first program module which processes query logs of the search engine to generate correlation data that reflects frequencies of occurrences of query terms within the same query. The system further comprises a second program module which uses at least the correlation data to sugguest refinements to search queries received from users.

The invention further includes a method of facilitating the refinement of search queries. The method comprises receiving a search query submitted by a user, and identifying a plurality of refined search queries, each of which comprises all terms of the query submitted by the user and an additional term. The method further comprises presenting each refined search query to the user as a respective link which is selectable to perform a corresponding search, such that the user may select a query refinement and initiate a refined search with a single selection action.

BRIEF DESCRIPTION OF THE DRAWINGS

These and other features will now be described with reference to the drawings summarized below. These drawings and the associated description are provided to illustrate a preferred embodiment of the invention, and not to limit the scope of the invention.

Throughout the drawings, reference numbers are re-used to indicate correspondence between referenced elements. In addition, the first digit of each reference number indicates the figure in which the element first appears.

FIG. 1 illustrates a system in which users access web site information via the Internet, and illustrates the basic web site components used to implement a search engine which operates in accordance with the invention.

FIG. 2 illustrates a sample book search page of the web site.

FIG. 3 illustrates sample log entries of a daily query log file.

FIG. 4 illustrates the process used to generate the correlation table of FIG. 1.

FIG. 5A illustrates a sample mapping before a query is added.

FIG. 5B illustrates a sample mapping after a query is added.

FIG. 6 illustrates a process for generating the correlation table from the most recent daily query log files.

FIG. 7 illustrates a process for selecting the related query terms from the correlation table.

FIG. 8A illustrates a set of related query terms from a single-term query.

FIG. 8B illustrates a set of intersecting terms and a set of related query terms from a multiple-term query.

FIG. 9 illustrates a sample search result page of the web site.

DETAILED DESCRIPTION OF THE PREFERRED EMBODIMENTS

The present invention provides a search refinement system and method for generating related query terms ("related terms") using a history of queries submitted to a search engine by a community of users. Briefly, the system generates query term correlation data which reflects the frequency with which specific terms have previously occurred together within the same query. The system uses the query term correlation data in combination with the query term(s) entered by the user to recommend additional query terms for refining the query. The incorporation of such historical query information into the process tends to produce related terms that are frequently used by other users in combination with the submitted query terms, and significantly increases the likelihood that these related terms will be helpful to the search refinement process. To further increase the likelihood that the related terms will be helpful, the correlation data is preferably generated only from those historical query submissions that produced a successful query result (at least one match).

In the preferred embodiment, the query term correlation date is regenerated periodically from recent query submissions, such as by using the last M days of entries in a query log, and thus heavily reflects the current tastes of users. As a result, the related terms suggested by the search engine tend to be terms that correspond to the most frequently searched items during the relevant time period. Thus, for example, in the context of a search engine of an online merchant, the search engine tends to suggest related terms that correspond to the current best-selling products. In one embodiment, the technique used to generate the related terms and present these terms to the user guarantees that the modified query will not produce a NULL query result.

The search refinement methods of the invention may be implemented, for example, as part of a web site, an Internet

site, an on-line services network, a document retrieval system, or any other type of computer system that provides searching capabilities to a community of users. In addition, the method may be combined with other methods for suggesting related terms, such as methods which process the contents of located documents.

A preferred web-based implementation of the search refinement system will now be described with reference to FIGS. 1–9. For purposes of illustration, the system is described herein in the context of a search engine that is used to assist customers of Amazon.com Inc. in locating items (e.g., books, CDs, etc.) from an on-line catalog of products. Throughout the description, reference will be made to various implementation-specific details of the Amazon.com implementation. These details are provided in order to fully illustrate a preferred embodiment of the invention, and not to limit the scope of the invention. The scope of the invention is set forth in the appended claims.

I. Overview of Web Site and Search Engine

FIG. **1** illustrates the Amazon.com web site **130**, including components used to implement a search engine in accordance with the invention.

As it is well known in the art of Internet commerce, the Amazon.com web site includes functionality for allowing users to search, browse, and make purchases from an on-line catalog of book titles, music titles, and other types of items via the Internet **120**. Because the catalog contains millions of items, it is important that the site provide an efficient mechanism for assisting users in locating items.

As illustrated by FIG. **1**, the web site **130** includes a web server application **131** ("web server") which processes user requests received from user computers **110** via the Internet **120**. These requests include queries submitted by users to search the on-line catalog for products. The web server **131** records the user transactions, including query submissions, within a query log **135**. In the embodiment depicted in FIG. **1**, the query log **135** consists of a sequence of daily query log files **135**(1)–**135**(M), each of which represents one day of transactions.

The web site **130** also includes a query server **132** which processes the queries by searching a bibliographic database **133**. The bibliographic database **133** includes information about the various products that users may purchase through the web site **130**. This information includes, for example, the titles, authors, publishers, subject descriptions, and ISBNs (International Standard Book Numbers) of book titles, and the titles, artists, labels, and music classifications of music titles. The information for each item is arranged within fields (such as an "author" field and a "title" field), enabling the bibliographic database **133** to be searched on a field-restricted basis. The site also includes a database **134** of HTML (Hypertext Markup Language) content which includes, among other things, product information pages which show and describe the various products.

The query server **132** includes a related term selection process **139** which identifies related query terms based on query term correlation data stored in a correlation table **137**. As depicted in FIG. **1** and described below, the correlation table **137** is generated periodically from the M most recent daily query log files **135**(1)–**135**(M) using an off-line table generation process **136**.

The web server **131**, query server **132**, table generation process **136**, and database software run on one or more Unix™-based servers and workstations (not shown) of the web site **130** although other types of platforms could be used. The correlation table **137** is preferably cached in RAM (random access memory) on the same physical machine as that used to implement the query server **132**. To accommodate large numbers of users, they query server **132** and the correlation table **137** can be replicated across multiple machines. The web site components that are invoked during the searching process are collectively referred to herein as a "search engine."

FIG. **2** illustrates the general format of a book search page **200** of the Amazon.com web site **130** that can be used to search the bibliographic database **133** for book titles. Users have access to other search pages that can be used to locate music titles and other types of products sold by the on-line merchant. The book search page **200** includes author, title, and subject fields **210**, **220**, **240** and associated controls that allow the user to initiate field-restricted searches for book titles. Users can perform searches by first typing in the desired information into a search field **210**, **220**, **240** and then clicking on the appropriate search button **230**, **250**. The term or string of terms submitted to the search engine is referred to herein as the "query." Other areas of the web site ask the user to submit queries without limiting the terms to specific fields.

When the user submits a query from the book search page **200** to the web site **130**, the query sever **132** applies the query to the bibliographic database **133**, taking into account any field restrictions within the query. If the query result is a single item, the item's product information page is presented to the user. If the query result includes multiple items, the list of items is presented to the user through a query result page which contains hypertextual links to the items' respective product information pages.

For multiple-term queries, the query server **132** effectively logically ANDs the query terms together to perform the search. For example, if the user enters the terms "JAVA" and "PROGRAMMING" into the title field **220**, the query server **132** will search for and return a list of all items that have both of these terms within the title. Thus, if any query term does not produce a match (referred to herein as a "non-matching term"), the query will produce a NULL query result. Presenting a NULL query result to the user can cause significant user frustration. To reduce this problem, in this event, the user may be presented with a list of items that are deemed to be "close matches." Although the search engine described herein logically ANDs the query terms together, it will be recognized that the invention can be applied to search engines that use other methods for processing queries.

In accordance with the invention, the search engine uses the query term correlation data stored in the correlation table **137** to select the related terms that best match the user's query. The search engine then presents the related terms to the user, allowing the user to refine the search and enhance discovery of relevant information. The query term correlation data indicates relationships between query terms, and is used to effectively predict query terms that are likely to be helpful to the search refinement process. In accordance with another aspect of the invention, the correlation table **137** preferably contains or reflects historical information about the frequencies with which specific query terms have appeared together within the same query.

The general format of the correlation table **137** is illustrated in FIG. **1**. In the embodiment depicted in FIG. **1** and described in detail herein, the correlations between query terms are based solely on frequency of occurrence within the same query. As described below, other types of query term correlations can additionally be used. In addition, although the disclosed implementation uses a table to store the query term correlation data, other types of databases can be used.

As illustrated by FIG. **1**, each entry within the correlation table **137** (two entries shown) has two primary components: a "key" term **140**, and a "related terms" list **142** for that key term. The related terms list **142** is a list of the N (e.g. 50) query terms that have appeared within the same query as the keyword with the highest degree of frequency, and is ordered according to frequency. For example, the entry for the key term COSMOS (ignoring the single-term prefixes, which are discussed below) is:

COSMOS: ASTRONOMY, SAGAN, UNIVERSE, . . .

indicating that ASTRONOMY has appeared together with COSMOS with the highest degree of frequency; SAGAN has appeared with COSMOS with the second highest degree of frequency, and so on. Each term that appears within the related terms list **142** is deemed to be related to the corresponding key term **140** by virtue of the relatively high frequency with which the terms have occurred within the same query.

As further depicted by FIG. **1**, each related term and each key term **140** preferably includes a single-character field prefix which indicates the search field **210**, **220**, **240** to which the term corresponds. These prefixes may, for example, be as follows: A=author, T=title, S=subject, R=artist, L=label, G=generic. In addition, each related term is stored together with a correlation score **146** which, in the preferred embodiment, indicates the number of times the related term has appeared in combination with the key term (within the search fields indicated by their respective field prefixes), not counting queries that produced a NULL query result.

Thus, for example, the related term (including prefix) S-ASTRONOMY has a correlation score of 410 under the key term T-COSMOS, indicating that four hundred and ten "successful" queries were received (during the time period to which the table **137** corresponds) which included the combination of COSMOS in the title field and ASTRONOMY in the subject field. Although the field prefixes and correlation scores **146** carry information which is useful to the related terms selection process (as described below), such information need not be preserved.

In operation, when a user submits a query to the web site **130**, the web server **131** passes the query to the query server **132**, and the query server applies the query to the bibliographic database **133**. If the number of items found exceeds a certain threshold (e.g., 50), the query server **132** invokes its related term selection process ("selection process") **139** to attempt to identify one or more related terms to suggest to the user. The selection process may alternatively be invoked without regard to whether a certain item count has been reached.

For each term in the query, the selection process **139** retrieves the respective related terms list **142** (if any) from the correlation table **137**, and if multiple lists result, merges these lists together. The selection process **139** then takes a predetermined number (e.g. 5) of the related terms from the top of the resulting list, and passes these "suggested terms" to the web server **131** with the query result listing. Finally, the web server **131** generates and returns to the user a query result page (FIG. **9**) which presents the suggested terms to the user for selection.

In one embodiment, the related terms lists are merged by retaining only the intersecting terms (terms which are common to all lists), and discarding all other terms. An important benefit of this method is that any single related term of the resulting list can be added to the query without producing a NULL query result. To take advantage of this feature, these related terms are preferably presented to the user using an interface method (as in FIG. **9**) which requires the user to add only one related term to the query per query submission.

The operation of the related term selection process **139** is described in further detail below.

The disclosed search engine also preferably uses historical query submissions and item selections to rank query results for presentation to the user. A preferred method for ranking query results based on such data is disclosed in U.S. patent application Ser. No. 09/041,081 filed Mar. 10, 1998. The search engine also preferably uses correlations between query terms to correct misspelled terms within search queries. A preferred method for correcting spelling errors in search queries is disclosed in U.S. patent application Ser. No. 09/115,662 entitled "System and Method for Correcting Spelling Errors in Search Queries," filed Jul. 15, 1998. The disclosures of these applications are hereby incorporated by reference.

II. Capturing and Processing of Query Information

As indicated above, the query term correlation data is preferably generated from the query log **135** using the table generation process ("generation process") **136**. In the preferred embodiment, the table generation process **136** is implemented as an off-line process which runs once a day and generates a new query correlation table **137**. The process effectively generates the table from the M most recent daily query log files **135**(1)–**135**(M). Using a relatively small M (e.g., 5) tends to produce query term correlation data that heavily reflects short term buying trends (e.g., new releases, weekly best-sellers, etc.), while using a larger M (e.g., 100) tends to produce a more comprehensive database. A hybrid approach can alternatively be used in which the table is generated from a large number of log files, but in which the most recent log files are given greater weight. For example, queries submitted during the last week can be counted three times when generating the correlation scores **146**, while queries submitted from one week to one month ago can be counted only once. In addition, rather than using M consecutive days of query submissions, the generation process **136** could use samples of query submissions from multiple different time periods.

In the preferred embodiment, the building of the query correlation table **137** consists of two primary phases: (1) generating daily log files, and (2) periodically parsing and processing these log files to generate the query correlation table **137**. Rather than generate new query term correlation data each time log information becomes available, the generation process **136** preferably generates and maintains separate query term correlation data for different constituent time periods of a relatively short length. In the preferred embodiment, the constituent time period is one day such that query term correlation data for a single day is stored in a daily results file. Each time query term correlation data is generated for a new constituent time period, the generation process **136** preferably combines this new data with existing data from earlier constituent time periods to form a collective query correlation table with information covering a longer composite period of time. This process is depicted in FIG. **6** and is described further below.

Any of a variety of alternative methods could be used to generate the correlation table **137**. For example, the generation process **136** could alternatively be implemented to update the query correlation table in real time by augmenting the table each time a user submits a successful query. In addition, the table generation process **136** and/or the selection process **139** could take into consideration other types of correlations between query terms, including extrinsic or "static" correlations that are not dependent upon the actions of users.

A. Generating Daily Query Log Files

A web server generally maintains a log file detailing all of the requests it has received from web browsers. The log file is generally organized chronologically and is made up of several entries, each containing information about a different request.

In accordance with the invention, each time a user performs a search, the web server **131** stores information about the submitted query in a log entry of a query log **135**. In addition, the web server **131** generates daily query log files **135(1)–135(M)** which each contain the log entries for a respective day. FIG. **3** illustrates four log entries of a sample daily query log file **135**. Each entry in the log file **135** includes information about a particular HTTP (Hypertext Transfer Protocol) transaction. The first log entry **310** contains date and time information for when the user submitted the query, the user identifier corresponding to the identity of the user (and, in some embodiments, identification of the particular interaction with the web server), the name of the web page where the query was entered, query terms entered by the user, and the number of the items found for the query. The "items_found" values in the log preferably indicate the number items that exactly matched the query.

For example, entry **310** indicates that at 2:23 AM on Feb. 13, 1998, user 29384719287 submitted the query {title= Snow Crash} from the book search page and that two items were found that exactly matched the query. Entry **320** indicates that the same user selected an item having an ISBN of 0553562614 about twenty seconds later, and that this selection was made from a search results page (as is evident from the HITP_REFERRER line). Other types of user actions, such as a request to place an item in a shopping cart or to purchase an item, are similarly reflected within the query log **135**. As indicated by the above example, a given users navigation path can be determined by comparing entries within the query log **135**.

B. Generating the Correlation Table

FIG. **4** shows the preferred method for generating the correlation table **137**. In tep **410** the generation process **136** goes through the most recent daily query log file to identify all multiple-term queries (i.e., queries comprised of more than one term) that returned at least one item ("items_ found">0) in the query result. In step **420**, the generation process **136** correlates each query ("key") term found in the set of queries to related terms that were used with the key term in a particular query, and assigns the related term a correlation score **146**. The correlation score indicates the frequency with which specific terms have historically appeared together within the same query during the period reflected by the daily query log. In step **430**, the generation process **136** stores the terms coupled with their correlation scores in a daily results file. In step **440**, the generation process **136** merges the daily results files for the last M days. Finally, in step **450**, the generation process **136** creates a new correlation table **137** and replaces the existing query correlation table.

In the preferred embodiment, the generation process **136** is executed once per day at midnight, just after the most recent daily query log is closed. In addition, it is assumed that the M−1 most recent daily query logs have already been processed by steps **410–430** of the process to generate respective daily results files.

Each of the steps **410–450** of the FIG. **4** process will now be described in greater detail.

Step 1: Processing the daily query log file

As indicated above, the generation process **136** parses the daily query log file in step **410** to identify and extract successful multi-term queries. Ignoring the query submissions that produced a NULL query result (items_found=0) provides the important benefits of (1) preventing non-matching terms from being added to the correlation table—either as keywords or as related terms—and (2) excluding potentially "weak" correlations between matching terms from consideration. In addition, as described below, excluding such "unsuccessful" query submissions enables the query terms selection process **139** to be implemented so as to guarantee that the modified query will produce a successful query result (i.e., a query result in which the item count is greater than zero).

Using the FIG. **3** log sequence as an example, the generation process **136** would parse the sample daily query log file **135** beginning with log entry **310**. The generation process **136** would extract the query for the first log entry **310** because the query contains more than one query term and "items_found" is greater than zero. Next, the generation process **136** would ignore entry **320** because it contains no query terms. The generation process **136** would then ignore entry **330** because although there are multiple query terms, the number of items found is not greater than zero. The generation process **136** would next extract the log entry **340** and continue through the daily query log file **135**. In some embodiments, other information such as query field or subsequent actions performed by the user may be used to determine which query submissions to extract or how heavily the queries should be weighted. In addition, other methods may be used to extract the information from the query log.

Step 2: Correlate terms

In accordance with the invention, the generation process **136** first takes each extracted query, and for each query term, adds a single-character field prefix ("prefix") which indicates the search field in which the query term was entered. Thus, for example, using the prefixes listed above, the prefix "T" would be added to the terms "SNOW" and "CRASH," in log entry **310**, and the prefix "S" would be added to the terms "OUTDOOR" and "TRAIL," in log entry **340**. During this process, identical terms that were submitted in different search fields are assigned different prefixes and are treated as different terms. For example, the term "SNOW" with a prefix of "T" would be treated as different from "SNOW" with the prefix "S." In the implementation described herein, the key term and related terms are stored without regard to alphabetic case, although case information can alternatively be preserved.

The generation process **136** then maps each query ("key") term found in the query and its prefix to other terms ("related terms") used with that particular query. A correlation score is maintained for each related term in the mapping based on the number of times the related term occurred in combination with the key term. The final values of the correlation scores taken over M days are stored within the query correlation table **137** as the correlation scores **146** depicted in FIG. **1**.

For example, if a user submits the query "ROUGH GUIDE TO LONDON," in the title field **220**, the terms would first be coupled with the prefix "T." The correlation scores in the mapping for "T-GUIDE," "T-TO," and "T-LONDON," relative to the key "T-ROUGH," would be incremented. Similarly, the correlation scores for the related terms under the keys "T-GUIDE," "T-TO," and "T-LONDON" would also be incremented.

FIG. **5A** illustrates an example mapping. In this figure, it is assumed that the generation process **136** has already processed many thousands of log entries. For each key term

140 stored in the table 137A, there is a related terms list 142 such that each related term in the list is coupled with a prefix and a value 146 representing the correlation score. Each time the key term 140 and a related term 142 are used together in a query, the related term's value 146 is incremented.

Assume that the table generation process 136 parses a query "OUTDOOR BIKE TRAIL" submitted in the subject field. FIG. 5A shows the mapping before the query is added. In response to the query, the generation process 136 updates the mapping 137A producing the mapping 137B shown in FIG. 5B. The generation process 136 first looks up the key term "S-OUTDOOR" 560 and then looks for the related terms "S-BIKE" 580 and "S-TRAIL" 590. If the related term is found, its value is incremented. If the related term is not found, the generation process 136 adds the related term and assigns it a beginning value. In the example shown in FIG. 5B, the values for both "S-BIKE" 580 and "S-TRAIL" 590 have been incremented by one. Note that under the key term "TOUTDOOR," the value for the term "S-TRAIL" was incremented while the value for the term "T-TRAIL" was not incremented. This is because the query was submitted in the subject field, thus affecting only terms with the prefix "S."

In some embodiments, certain key terms may be excluded from the mapping if they are frequently used, and yet do not further the search refinement process. For example, common articles such as "THE," "A," "TO," and "OF" may be excluded from the mapping. While only three partial entries are depicted in FIG. 5A, many thousands of entries would be stored in a typical daily results file. In the preferred implementation, the mapping for a daily query log file is stored in a B-tree data structure. In other embodiments, a linked list, database, or other type of data structure can be used in place of the B-tree.

In addition, the amount by which the correlation scores are incremented may be increased or decreased depending on different kinds of selection actions performed by the users on items identified in query results. These may include whether the user displayed additional information about an item, how much time the user spent viewing the additional information about the item, how many hyperlinks the user followed within the additional information about the item, whether the user added the item to his or her shopping basket, and whether the user ultimately purchased the item. For example, a given query submission can be counted twice (such as by incrementing the correlation score by two) if the user subsequently selected an item from the query result page, and counted a third time if the user then purchased the item or added the item to the shopping basket. These and other types of post-search activities reflect the usefulness of the query result, and can be extracted from the query log 135 using well-known tracing methods.

Step 3: Create Daily Results File

Once the mapping is complete, that is, all entries in the daily query log file have been parsed, the generation process 136 creates a daily results file (step 430) to store the B-tree. In other embodiments, the daily results file may be generated at an earlier stage of the process, and may be incrementally updated as the parsing occurs.

Step 4: Merge Daily Results Files

In step 440, the generation process 136 generates the query correlation table 137 for a composite period by combining the entries of the daily results files for the length of the composite period. As depicted in FIG. 6, the table generation process 136 regenerates the query correlation table 137 on a daily basis from the M most recent daily results files, where M is a fixed number such as 10 or 20. Each day, the daily results file created in step 430 is merged with the last M−1 daily results files to produce the query correlation table 137

For example, in FIG. 6, suppose the generation process 136 generates a daily results file for Feb. 7, 1998 610 and is set to generate a new query correlation table for the period of the last seven days (M=7). At the end of Feb. 7, 1998, the generation process 136 would merge the daily results files from the past seven days for the composite period of Feb. 1, 1998 to Feb. 7, 1998 to form a new query correlation table 137A. At the end of Feb. 7, 1998, the generation process 136 would generate a daily results file for Feb. 8, 1998 630 and then merge the daily results files from the past seven days for the composite period of Feb. 2, 1998 to Feb. 8, 1998 to form a new query correlation table 137B. When the entries are merged, the scores of the corresponding entries are combined, for example, by summing them. In one embodiment, the scores in more recent daily results files are weighted more heavily than those scores in less recent daily results files, so that the query term correlation data more heavily reflects recent query submissions over older query submissions. This "sliding window" approach advantageously produces a query correlation table that is based only on recent query submissions, and which thus reflects the current preferences of users. For example, if a relatively large number of users have searched for the book Into Thin Air by Jon Krakauer over the past week, the correlations between the terms "T-INTO," "T-THIN," "T-AIR," and "A-KRAKAUER" will likely be correspondingly high; a query which consists of a subset of these terms will thus tend to produce a related terms lists which includes the other terms.

Step 5: Replace Old Query Correlation Table With New Query Correlation Table

In step 450, once the daily results files have been merged, the generation process 136 sorts the related terms lists from highest-to-lowest score. The generation process 136 then truncates the related terms lists to a fixed length N (e.g., 50) and stores the query correlation table in a B-tree for efficient lookup. The new query correlation table 137 B-tree is then cached in RAM (random access memory) in place of the existing query correlation table.

III. Using the Table to Generate Related terms

As indicated above, the query server 132 uses the query correlation table 137 to select related terms to be suggested to the user. More specifically, when a user performs a search which identifies more than a predetermined number of items, the related term selection process ("selection process") 139 returns a query result listing items that match the query along with a set of related terms generated from the query correlation table. An important benefit of this method is that it is highly efficient, allowing the query result page to be returned without adding appreciable delay. Further, the small delay added by the related terms selection process can be completely avoided by optionally generating the related terms concurrently with the search of the bibliographic database 133 (rather than waiting to see if a threshold item count is reached).

FIG. 7 illustrates the sequence of steps performed by the selection process 139. The selection process 139 first enters a loop (steps 710–740) in which the selection process 139 looks up a query term in the correlation table and then retrieves the term's related terms list 142. This continues for each term in the query. Next, if the query has multiple terms, in step 760, the selection process 139 combines the related terms lists. The lists are preferably combined by taking the intersection of the related terms lists (i.e., deleting terms

which do not appear in all lists) and summing the correlation scores of the remaining terms. At this point, every term which remains in the list is a term which has appeared, in at least one prior, successful query, in combination with every term of the present query. Thus, assuming entries have not been deleted from the bibliographic database **133** since the beginning of the composite time period (the period to which the table **137** applies), any of these terms can be added individually to the present query without producing a NULL query result. In other embodiments, the selection process **139** combines the related terms lists by summing the correlation scores of terms common to other related terms lists, without deleting any terms. Another implementation might give weighted scores for intersecting terms such that terms appearing in more than one related terms list are weighted heavier than those terms appearing only in a single related terms list.

In step **770**, the selection process **139** selects the X terms with the highest values from the list, where X can be any desired number. In one embodiment, the selection process **139** chooses the top X related terms without regard to the field prefixes of these related terms. The selection process may alternatively be configured to select only those related terms that correspond to the search field(s) of the present query; for example, if the query was entered into the subject field **240** (FIG. **2**), the user may be presented only with other subject terms (related terms with the prefix "S").

For single-term queries, the selection process **139** thus retrieves the top X terms from the table. FIG. **8**A illustrates the related terms that would be generated for a single-term query of "TRAIL" in the subject field using the mapping from FIG. **5**B. The selection process **139** would look up the key term "S-TRAIL" **570** and select X related terms with the highest X values. For example, suppose the selection process **139** were configured to suggest three related terms (X=3) that correspond to the search field(s) of the present query. The selection process **139** would then look up the key term "S-TRAIL" **570** and display the three related terms with the top three values **810** and with the same prefix as the key term, as illustrated in FIG. **8**A.

For multiple-term queries, the selection process **139** obtains the related terms lists **142** for each of the query terms, and then takes the intersection of these lists. FIG. **8**B illustrates the related term results for a multiple-term query in the subject field of "OUTDOOR TRAIL" using the mapping from FIG. **5**B. The selection process **139** would look up the key terms "S-OUTDOOR" **560** and "S-TRAIL" **570** and see if they have any related terms in common. In the mapping, the related terms "S-BIKE," "S-SPORTS," and "S-VACATION" are found under the key terms "S-OUTDOOR" **560** and "S-TRAIL," **570**; thus "S-BIKE," "S-SPORTS," and "S-VACATION" are the intersecting terms **820** as illustrated in FIG. **8**B. The selection process **139** would then display the X intersecting terms with the same prefix and the X highest summed correlation scores. If there were less than X intersecting, related terms, the selection process **139** could show the intersecting terms with any prefix or use other criteria to generate the remaining related terms. For example, the process **139** could take the top Y terms with the highest summed correlation scores from the non-intersecting related terms, although suggesting such terms could produce a NULL query result.

As indicated above, the method can alternatively be implemented without preserving or taking into account search field information. In addition, the method can be appropriately combined with other techniques for generating related terms, including techniques which use the contents of the query result.

IV. Presenting the Related Query terms to the User

There are a number of different ways to present the related terms to the user, including the conventional methods (check boxes and drop-down menus) described above. In implementations which suggest only the intersecting related terms, an interface which requires the user to add no more than one related term per query submission is preferably used, so that the modified query will not produce a NULL query result.

In the preferred embodiment, the related terms are presented though hypertextual links which combine both the original query term(s) and a respective related term. For example, if the user enters the query "ROUGH" in the subject field, three additional hyperlink may be displayed on the query result page, each of which generates a modified search when clicked on by the user. Each of these links is formed by combining the user's query with a related term (e.g., the three hyperlinks might be "ROUGH—GUIDE," "ROUGH—LONDON," and "ROUGH—TERRAIN"). When the user clicks on one of these links, the corresponding modified query is submitted to the search engine. The method thus enables the user to select and submit the modified query with a single action (e.g., one click of a mouse). As an inherent benefit of the above-described method of generating the related terms, each such link produces as least one "hit."

FIG. **9** illustrates a sample query result page **900** in which a user has performed a subject field search on the terms "OUTDOOR TRAIL" and has received a set of three related terms, each of which is incorporated into a respective hyperlink **910**. The page will also typically contain a listing of the query result items **920**. If the user clicks on the hyperlink "OUTDOOR TRAIL—BIKE," the search engine will perform a search using the terms "S-OUTDOOR," "S-TRAIL," and "S-BIKE," and will then return the associated items. The query result page **900** may also have search fields (not shown) for allowing the user to edit the query.

Any of a variety of additional techniques can be used in combination with this hyperlink-based interface. For example, in one embodiment, the query server **132** automatically selects the related term at the top of related terms list (such as the term "bike" in the FIG. **9** example), and searches the query result to identify a subset of query result items that include this related term. The query server **132** thereby effectively applies the "top" suggested modified query to the bibliographic database **133**. This process could be repeated using additional related terms in the list. The items within the subset can then be displayed to the user at the top of the query result list, and/or can be displayed in highlighted form. Further, the query server **132** could cache the list of items that fall within the subset, so that if the user submits the modified query (such as by clicking on the link "OUTDOOR BIKE—TRAIL" in FIG. **9**), the query server could return the result of the modified search without having to search the bibliographic database. Special tags or codes could be embedded within the modified-query hyperlinks and passed to the web site **130** to enable the query server **132** to match the modified queries to the cached results.

Although this invention has been described in terms of certain preferred embodiments, other embodiments that are apparent to those of ordinary skill in the art are also within the scope of this invention. Accordingly, the scope of the present invention is defined only by reference to the appended claims.

In the claims which follow, reference characters used to denote process steps are provided for convenience of description only, and not to imply a particular order for performing the steps.

What is claimed is:

1. In a computer system that implements a search engine that is accessible to a community of users, a method of assisting users in refining search queries, the method comprising:

receiving a search query submitted by a user, the search query comprising at least one term;

using a history of search queries submitted to the search engine over a selected period of time by the community of users to identify at least one refinement to the search query; and

suggesting the at least one refinement to the user.

2. The method as in claim 1, wherein identifying the at least one refinement comprises using the history of search queries to identify an additional term that has appeared in combination with each term of the query submitted by the user relatively frequently over the selected period of time.

3. The method as in claim 1, wherein suggesting the at least one refinement comprises presenting a plurality of augmented search queries to the user as respective hyperlinks that are selectable by the user to initiate corresponding searches.

4. The method as in claim 1, wherein identifying the at least one refinement comprises using the history of search queries to determine that none of the at least one refinement does produces a NULL search result.

5. The method as in claim 1, wherein the period of time is selected such that recent historical search queries are given more weight than aged historical search queries, so that suggested refinements tend to reflect current interests of the community of users.

6. A system for assisting users in refining search queries submitted to a search engine, comprising:

a first program module which processes query logs of the search engine to generate correlation data that reflects frequencies of occurrences of query terms within the same query; and

a second program module which uses at least the correlation data to suggest refinements to search queries received from users.

7. The system as in claim 6, wherein the first program module ignores query submissions that produced NULL search results.

8. The system as in claim 6, wherein the first program module generates the correlation data periodically from a most recent set of historical query submissions, so that refinements suggested by the second program module reflect current interests of users.

9. The system as in claim 6, wherein the second program module presents refinements to users using a user interface in which each suggested refinement is presented as a respective link that is selectable to initiate a refined search.

10. A method of facilitating refinement of search queries, comprising:

receiving a search query submitted by a user;

identifying a plurality of refined search queries, each of which comprises all terms of the query submitted by the user and an additional term; and

presenting each refined search query to the user as a respective link which is selectable to perform a corresponding search;

wherein the method allows the user to select a query refinement and initiate a refined search with a single selection action.

11. The method as in claim 10, wherein identifying a plurality of refined search queries comprises using historical query data to identify refined queries that reflect current interests of users.

12. The method as in claim 10, wherein identifying a plurality of refined search queries comprises using a history of query submissions to determine that none of the refined search queries produces a NULL search result.

* * * * *

EXHIBIT 3

US006266649B1

(12) **United States Patent**
Linden et al.

(10) **Patent No.: US 6,266,649 B1**
(45) **Date of Patent: Jul. 24, 2001**

(54) **COLLABORATIVE RECOMMENDATIONS USING ITEM-TO-ITEM SIMILARITY MAPPINGS**

(75) Inventors: **Gregory D. Linden; Jennifer A. Jacobi; Eric A. Benson**, all of Seattle, WA (US)

(73) Assignee: **Amazon.Com, Inc.**, Seattle, WA (US)

(*) Notice: Subject to any disclaimer, the term of this patent is extended or adjusted under 35 U.S.C. 154(b) by 0 days.

(21) Appl. No.: **09/157,198**

(22) Filed: **Sep. 18, 1998**

(51) **Int. Cl.**[7] **G06F 17/60**

(52) **U.S. Cl.** **705/26**; 705/27; 705/14

(58) **Field of Search** 705/26, 27, 14; 235/383; 380/24; 455/5.1, 450; 707/3, 102; 709/227

(56) **References Cited**

U.S. PATENT DOCUMENTS

4,870,579	*	9/1989	Hey	364/419
4,992,940	*	2/1991	Dworkin	364/401
4,996,642	*	2/1991	Hey	364/419

(List continued on next page.)

FOREIGN PATENT DOCUMENTS

0 265 083	*	4/1988	(EP)	G09F/27/00
0751471	*	1/1997	(EP)	G06F/17/60
0 827 063 A1		3/1998	(EP) .	
2 336 925	*	4/1988	(GB)	G06F/17/00

OTHER PUBLICATIONS

Joaquin Delgado, "Intelligent Collaborative Information Retrieval".*
Joaquin Delgado, "Content-based Collaborative Information Filtering".*
Marko Balabanovic and Yoav Shoham, "Content-Based, Collaborative Recommendation," Communications of the ACM, v 40n3, pp. 66–72, Mar. 1997.*

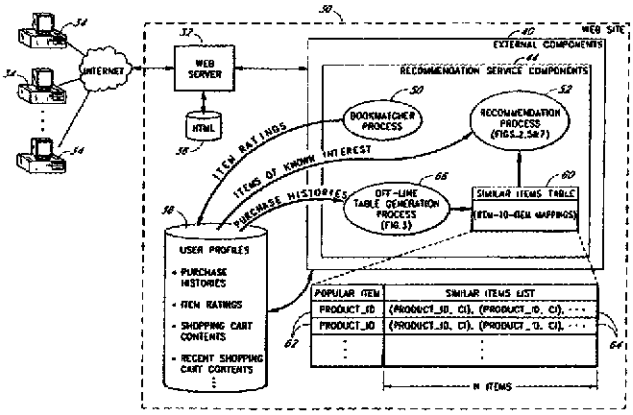
(List continued on next page.)

Primary Examiner—Tariq R. Hafiz
Assistant Examiner—J Harle
(74) *Attorney, Agent, or Firm*—Knobbe, Martens, Olson & Bear, LLP

(57) **ABSTRACT**

A recommendations service recommends items to individual users based on a set of items that are known to be of interest to the user, such as a set of items previously purchased by the user. In the disclosed embodiments, the service is used to recommend products to users of a merchant's Web site. The service generates the recommendations using a previously-generated table which maps items to lists of "similar" items. The similarities reflected by the table are based on the collective interests of the community of users. For example, in one embodiment, the similarities are based on correlations between the purchases of items by users (e.g., items A and B are similar because a relatively large portion of the users that purchased item A also bought item B). The table also includes scores which indicate degrees of similarity between individual items. To generate personal recommendations, the service retrieves from the table the similar items lists corresponding to the items known to be of interest to the user. These similar items lists are appropriately combined into a single list, which is then sorted (based on combined similarity scores) and filtered to generate a list of recommended items. Also disclosed are various methods for using the current and/or past contents of a user's electronic shopping cart to generate recommendations. In one embodiment, the user can create multiple shopping carts, and can use the recommendation service to obtain recommendations that are specific to a designated shopping cart. In another embodiment, the recommendations are generated based on the current contents of a user's shopping cart, so that the recommendations tend to correspond to the current shopping task being performed by the user.

49 Claims, 7 Drawing Sheets



U.S. PATENT DOCUMENTS

5,235,509	*	8/1993	Mueller et al.	364/405
5,583,763	*	12/1996	Atcheson et al.	364/551.01
5,704,017	*	12/1997	Heckerman et al.	395/61
5,749,081	*	5/1998	Whiteis	707/102
5,774,670	*	6/1998	Montulli	395/200.57
5,790,426	*	8/1998	Robinson	364/554
5,794,210		8/1998	Goldhaber et al.	
5,867,799	*	2/1999	Lang et al.	707/1
5,884,282	*	3/1999	Robinson	705/27
5,905,973	*	5/1999	Yonezawa et al.	705/27
5,909,023	*	6/1999	Ono et al.	235/380
5,909,492	*	6/1999	Payne et al.	380/24
5,918,014	*	6/1999	Robinson	395/200.49
6,006,218	*	12/1999	Breese et al.	707/3
6,018,738	*	1/2000	Breese et al.	707/100

OTHER PUBLICATIONS

"Net Perceptions Closes Second Round of Financing: GroupLens secures No. 1 recommendation system spot with strong endorsement by investment community", Business Wire, p.3020013, Dialog File 16, AN 05495619, Mar. 1998.*

"LinkShare Launches Affiliates Profiling Software; First to Integrate Personalization Software Into Affiliates Program", PR Newswire, LinkShare Corp., Dialog File 813 AN 1232636, Feb. 1998.*

"Fort Point Partners Teams With LikeMinds to Offer Breakthrough Personalization Technology for Increased Sales Online", Business Wire, p.3110064, Dialog File 16, AN 05510541, Mar. 1998.*

"Net Perceptions Debuts GroupLens Version 3.0 at Internet World Spring; 'Industrial Strength Tool Matures Into Essential Website Technology'", Business Wire, p. 3090007, Dialog File 16, AN 05505690, Mar. 1998.*

"Home Box Office Selects Like Minds Personalization Software for Second Network Site", PR Newswire, p. 1117SFM023, Dialog File 148, AN 09869396, Nov. 1997.*

"GroupLens Recommendation Engine to Standardize Internet Personalization For Singapore's Online Technologies Consortium", Business Wire, Dialog File 20, AN 01951318, Jun. 1998.*

Borchers, A. et al., "Ganging up on Information Overload", Computer, pp. 106–108, Apr. 1998.*

Konstan, J. et al., "GroupLens: Applying Collaborative Filtering to Usenet News", Communications of the ACM, vol. 30, No. 3, pp. 77–87, Mar. 1997.*

Miller, B. et al., "Experiences with GroupLens: Making Usenet Useful Again", 1997 Annual Technical Conference, pp. 219–232, 1997.*

Resnick, P. et al., "Recommender Systems", Communications of the ACM, vol. 40, No. 3, pp. 56–58, Mar. 1997.*

Rucker J. et al., "Siteseer: Personalized Navigation for the Web", Communications of the ACM, vol. 40, No. 3, pp. 73–76, Mar. 1997.*

Brier, S.E., "Smart Devices Peep Into Your Grocery Cart", New York Times Co., Section G, p. 3, Col. 3, Circuits, Jul. 1998.*

"COSMOCOM", Computer Telephony, p. 124, Jul. 1998.*

Nash, E.L., "Direct Marketing; Strategy, Planning, Execution", 3rd Ed., McGraw–Hill, Inc., pp. 165 & 365–6, 1994.*

"iCat Electronic Commerce Suite Takes 'Best of Show' Award at WebInnovation 97", PR Newswire. Jun. 1997.*

"ICAT Corporation: iCat's Commerce Suite Makes Setting Up Shop on Net Even Easier Than High Street", M2 Presswire, Feb. 1997.*

Dragan et al., "Advice From the Web", PC Magazine, vol. 16, No. 15, p. 133, Sep. 1997.*

"Able Solutions Announces Able Commerce 2.6", PR Newswire, Sep. 1998.*

"Internet World—IBM To Expand E–Comm Features", Newsbytes News Network, Dec. 1996.*

McMains, A., "Weiss, Whitten, Staliano's", ADWEEK Eastern Edition, vol. 39, No. 24, p. 82, Jun. 1998.*

"Cdnow Rated Top Music Site by eMarketer, the Authority on Business Online", PR Newswire, Sep. 1998.*

Upendra Shardanand and Pattie Maes with MIT Media–Lab, Social Information Filtering: Algorithms for Automating "Word of Mouth", 8 pgs (undated).

Combining Social Networks and Collaborative Filtering, Communications of the ACM, Mar. 1997/vol. 40, No. 3, pp. 63–65.

Pointing the Way: Active Collaborative Filtering, CHI '95 Proceedings Papers, 11 pgs.

Bradley N. Miller, John T. Riedl, Joseph A. Konstan with Department of Computer Science, University of Minnesota, Experiences with GroupLens: Making Usenet Useful Again, 13 pgs (undated).

A System for Sharing Recommendations, Communications of the ACM, Mar. 1997/vol. 40, No. 3, pp. 59–62.

Recommender Systems for Evaluating Computer Messages, Communications of the ACM, Mar. 1997/vol. 40, No. 3, pp. 88 and 89.

Content–Based, Collaborative Recommendation, Communications of the ACM, Mar. 1997/vol. 40, No. 3, pp. 66–72.

Applying Collaborative Filtering to Usenet News, Communications of the ACM, Mar. 1997/vol. 40, No. 3, pp. 77–87.

Personalized Navigation for the Web, Communications of the ACM, Mar. 1997/vol. 40, No. 3, pp. 73–76.

GroupLens: An Open Architecture for Collaborative Filtering of Netnews, 18 pgs.

Net Perceptions, Inc., White Paper, Building Customer Loyalty and High–Yield Relationships Through GroupLens Collaborative Filtering, 9 pgs., Nov. 22, 1996.

Christos Faloutsos and Douglas Oard with University of Maryland, A Survey of Information Retrieval and Filtering Methods, 22 pgs. (undated).

* cited by examiner

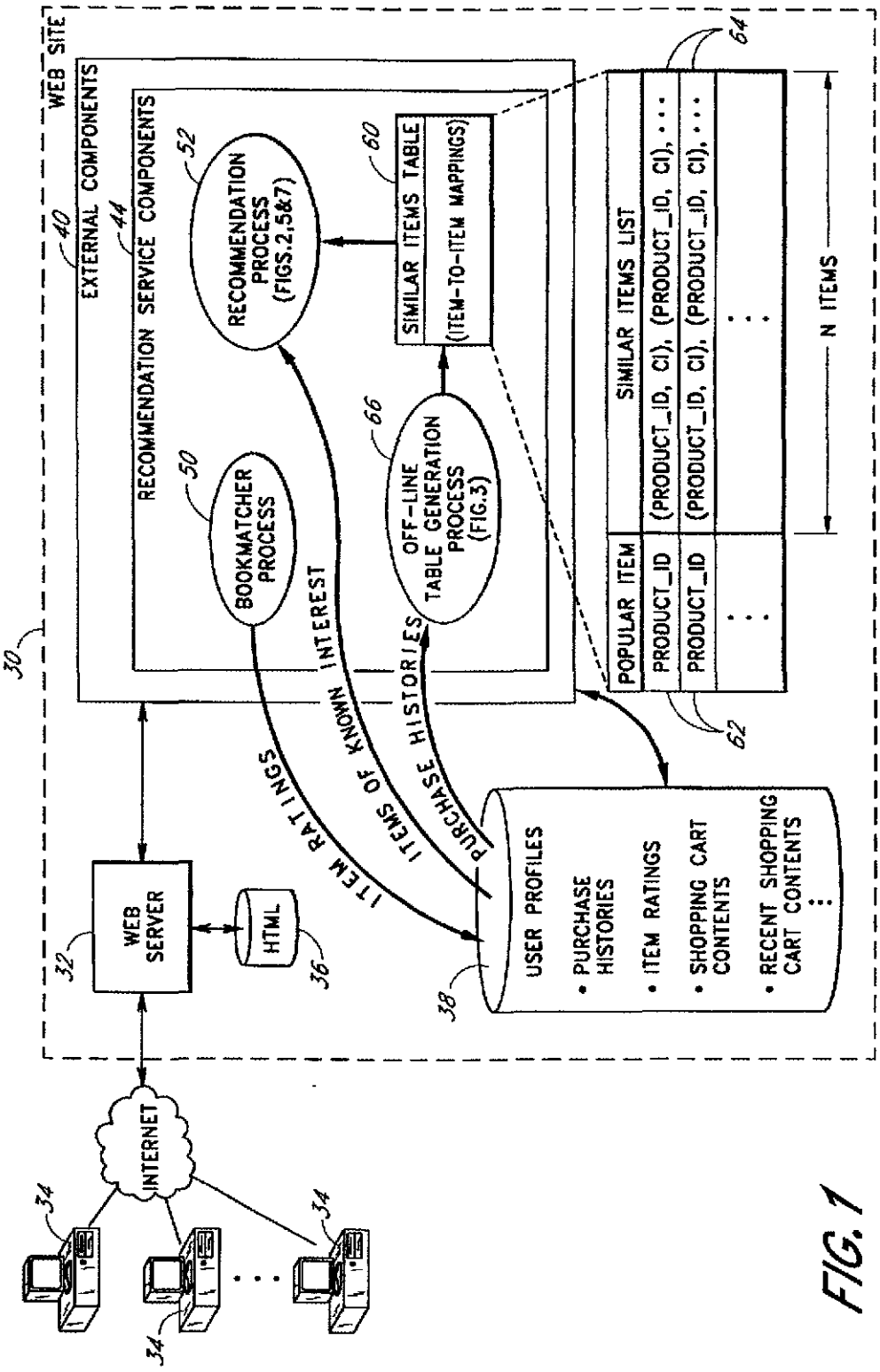

WEB SITE
30
EXTERNAL COMPONENTS
40
RECOMMENDATION SERVICE COMPONENTS
44
RECOMMENDATION PROCESS (FIGS.2,5&7)
52
BOOKMATCHER PROCESS
50
OFF-LINE TABLE GENERATION PROCESS (FIG.3)
66
SIMILAR ITEMS TABLE
(ITEM-TO-ITEM MAPPINGS)
60
ITEM RATINGS
ITEMS OF KNOWN INTEREST
PURCHASE HISTORIES
WEB SERVER
32
HTML
36
USER PROFILES
• PURCHASE HISTORIES
• ITEM RATINGS
• SHOPPING CART CONTENTS
• RECENT SHOPPING CART CONTENTS
38
POPULAR ITEM
SIMILAR ITEMS LIST
PRODUCT_ID
(PRODUCT_ID, C1), (PRODUCT_ID, C1), . . .
PRODUCT_ID
(PRODUCT_ID, C1), (PRODUCT_ID, C1), . . .
62
64
N ITEMS
INTERNET
34
34
34

FIG. 1

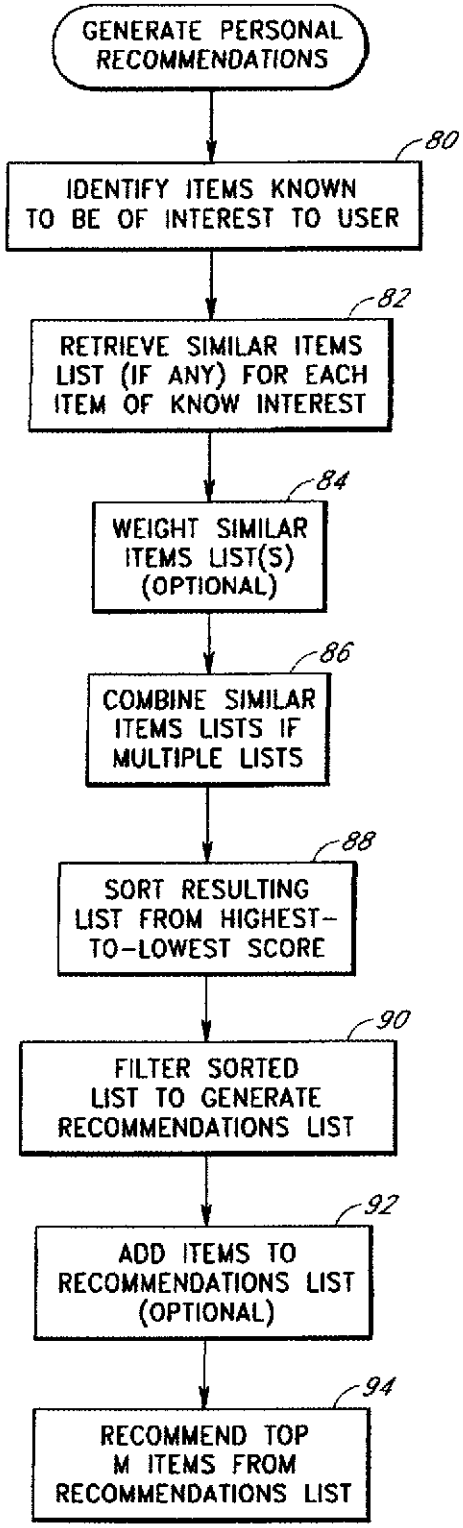

GENERATE PERSONAL RECOMMENDATIONS
80
IDENTIFY ITEMS KNOWN TO BE OF INTEREST TO USER
82
RETRIEVE SIMILAR ITEMS LIST (IF ANY) FOR EACH ITEM OF KNOW INTEREST
84
WEIGHT SIMILAR ITEMS LIST(S) (OPTIONAL)
86
COMBINE SIMILAR ITEMS LISTS IF MULTIPLE LISTS
88
SORT RESULTING LIST FROM HIGHEST-TO-LOWEST SCORE
90
FILTER SORTED LIST TO GENERATE RECOMMENDATIONS LIST
92
ADD ITEMS TO RECOMMENDATIONS LIST (OPTIONAL)
94
RECOMMEND TOP M ITEMS FROM RECOMMENDATIONS LIST

FIG. 2

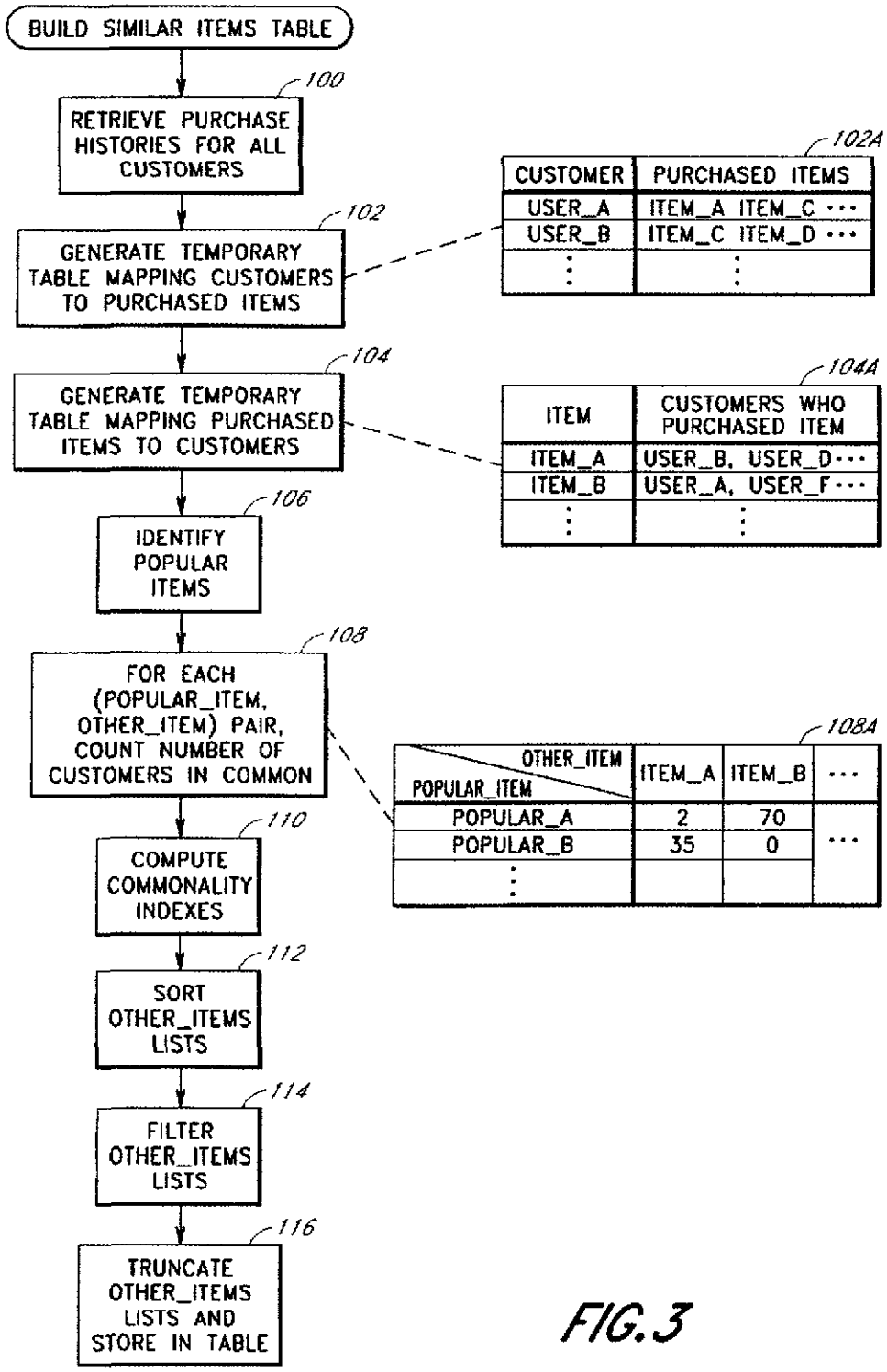

BUILD SIMILAR ITEMS TABLE
100
RETRIEVE PURCHASE HISTORIES FOR ALL CUSTOMERS
102
GENERATE TEMPORARY TABLE MAPPING CUSTOMERS TO PURCHASED ITEMS
102A
CUSTOMER
PURCHASED ITEMS
USER_A
ITEM_A ITEM_C ···
USER_B
ITEM_C ITEM_D ···
104
GENERATE TEMPORARY TABLE MAPPING PURCHASED ITEMS TO CUSTOMERS
104A
ITEM
CUSTOMERS WHO PURCHASED ITEM
ITEM_A
USER_B, USER_D···
ITEM_B
USER_A, USER_F···
106
IDENTIFY POPULAR ITEMS
108
FOR EACH (POPULAR_ITEM, OTHER_ITEM) PAIR, COUNT NUMBER OF CUSTOMERS IN COMMON
108A
OTHER_ITEM
POPULAR_ITEM
ITEM_A
ITEM_B
···
POPULAR_A
2
70
POPULAR_B
35
0
110
COMPUTE COMMONALITY INDEXES
112
SORT OTHER_ITEMS LISTS
114
FILTER OTHER_ITEMS LISTS
116
TRUNCATE OTHER_ITEMS LISTS AND STORE IN TABLE

FIG. 3

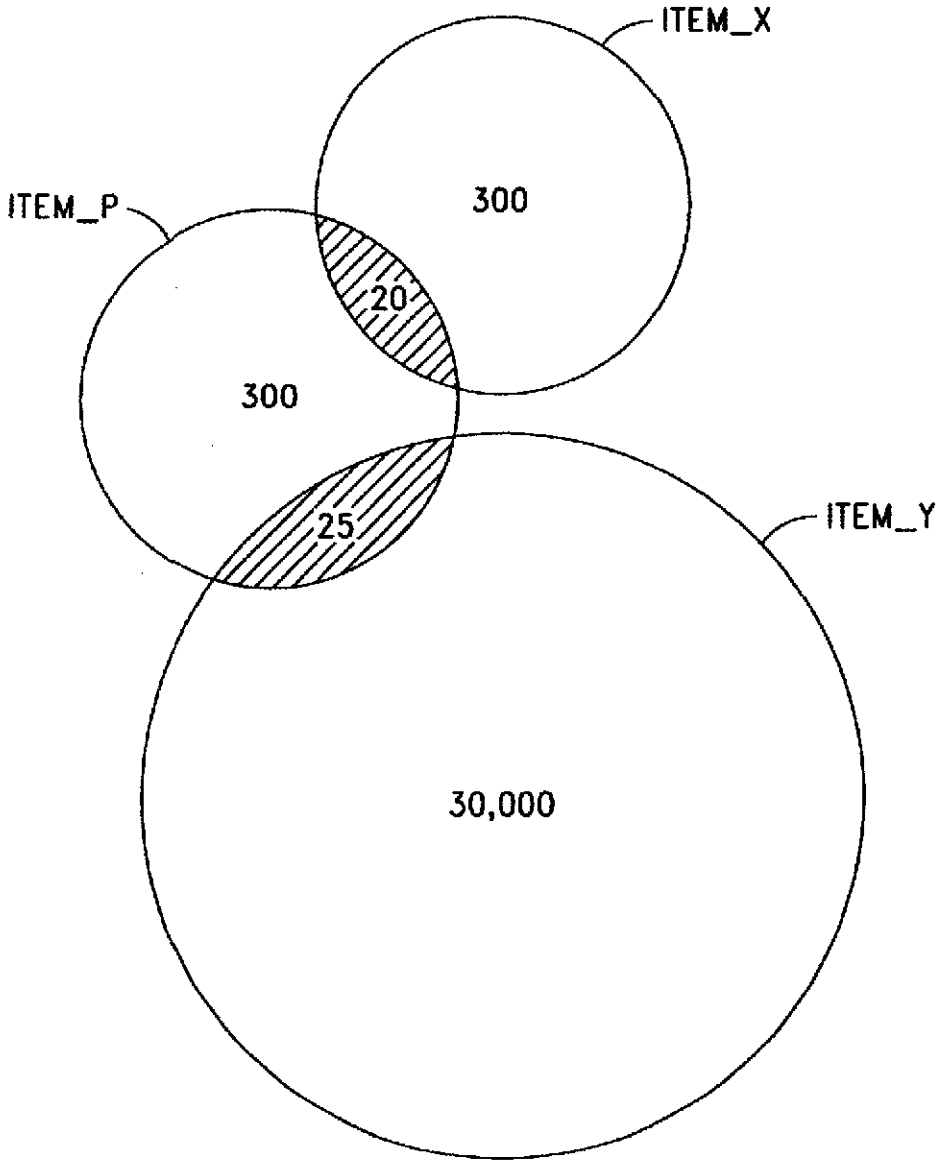


FIG. 4

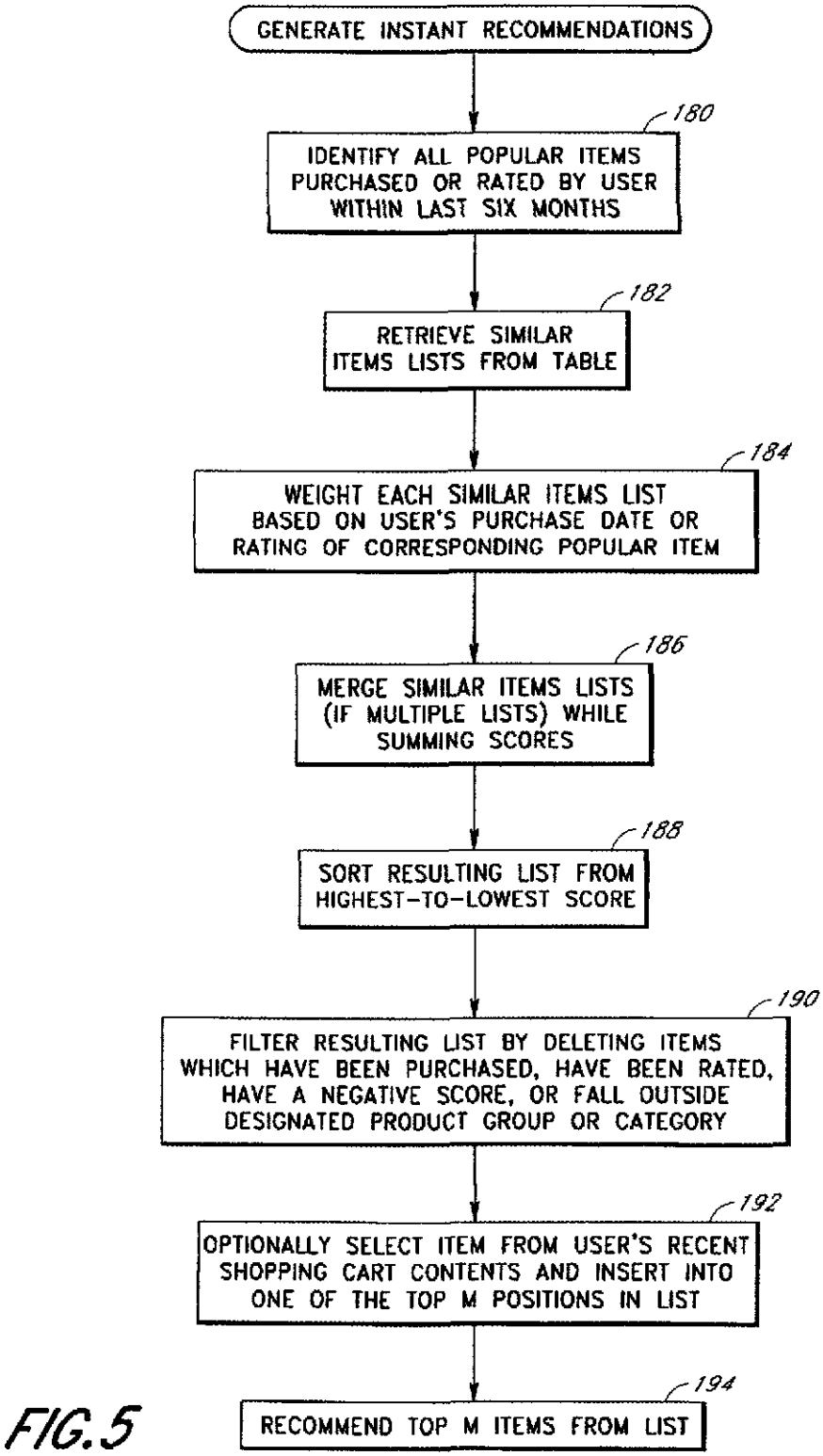

GENERATE INSTANT RECOMMENDATIONS
180
IDENTIFY ALL POPULAR ITEMS PURCHASED OR RATED BY USER WITHIN LAST SIX MONTHS
182
RETRIEVE SIMILAR ITEMS LISTS FROM TABLE
184
WEIGHT EACH SIMILAR ITEMS LIST BASED ON USER'S PURCHASE DATE OR RATING OF CORRESPONDING POPULAR ITEM
186
MERGE SIMILAR ITEMS LISTS (IF MULTIPLE LISTS) WHILE SUMMING SCORES
188
SORT RESULTING LIST FROM HIGHEST-TO-LOWEST SCORE
190
FILTER RESULTING LIST BY DELETING ITEMS WHICH HAVE BEEN PURCHASED, HAVE BEEN RATED, HAVE A NEGATIVE SCORE, OR FALL OUTSIDE DESIGNATED PRODUCT GROUP OR CATEGORY
192
OPTIONALLY SELECT ITEM FROM USER'S RECENT SHOPPING CART CONTENTS AND INSERT INTO ONE OF THE TOP M POSITIONS IN LIST
194
RECOMMEND TOP M ITEMS FROM LIST

FIG. 5

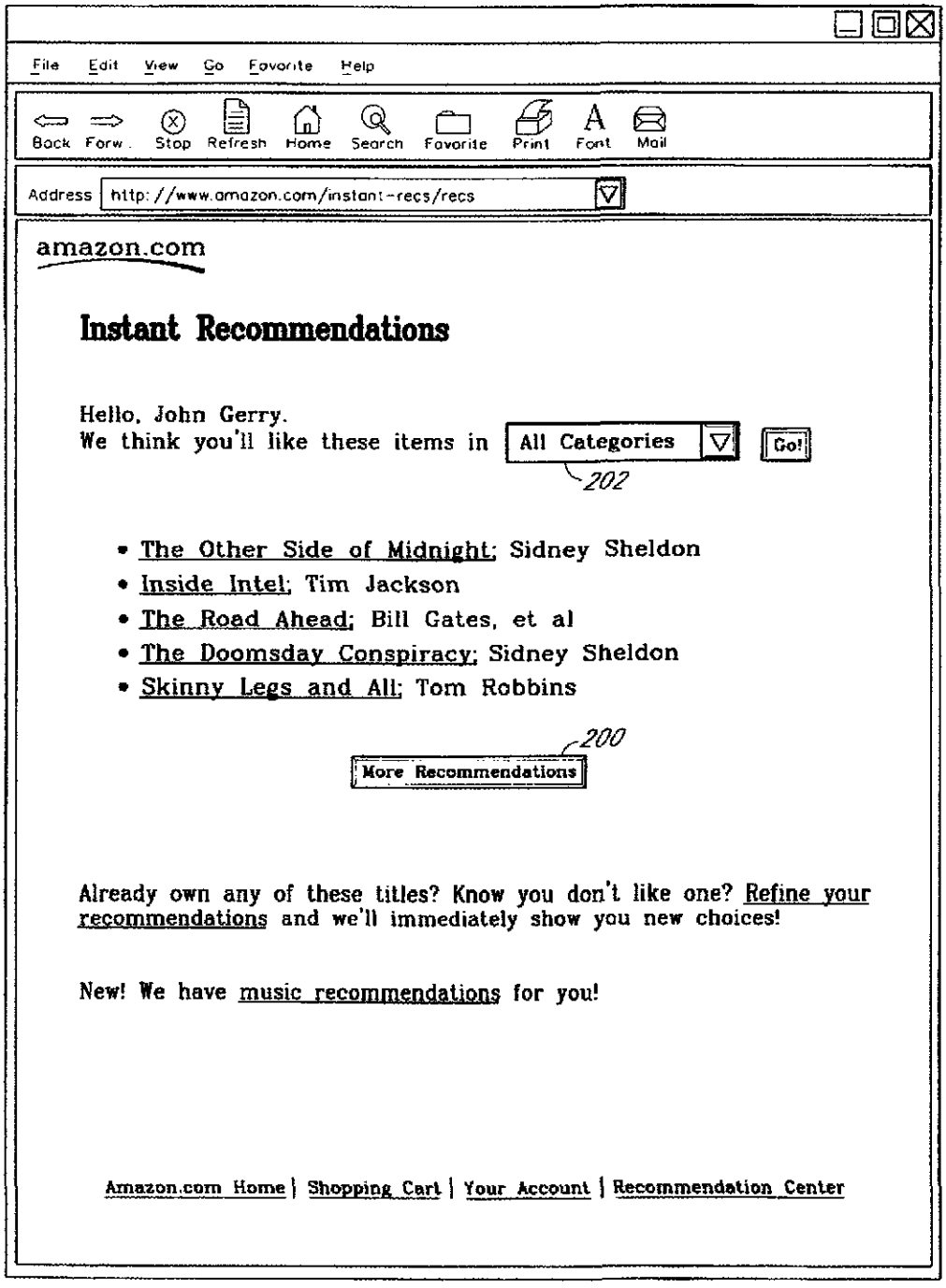

File Edit View Go Favorite Help
Back Forw. Stop Refresh Home Search Favorite Print Font Mail
Address http://www.amazon.com/instant-recs/recs
amazon.com
Instant Recommendations
Hello, John Gerry.
We think you'll like these items in All Categories Go!
202
The Other Side of Midnight; Sidney Sheldon
Inside Intel; Tim Jackson
The Road Ahead; Bill Gates, et al
The Doomsday Conspiracy; Sidney Sheldon
Skinny Legs and All; Tom Robbins
200
More Recommendations
Already own any of these titles? Know you don't like one? Refine your recommendations and we'll immediately show you new choices!
New! We have music recommendations for you!
Amazon.com Home | Shopping Cart | Your Account | Recommendation Center

FIG. 6

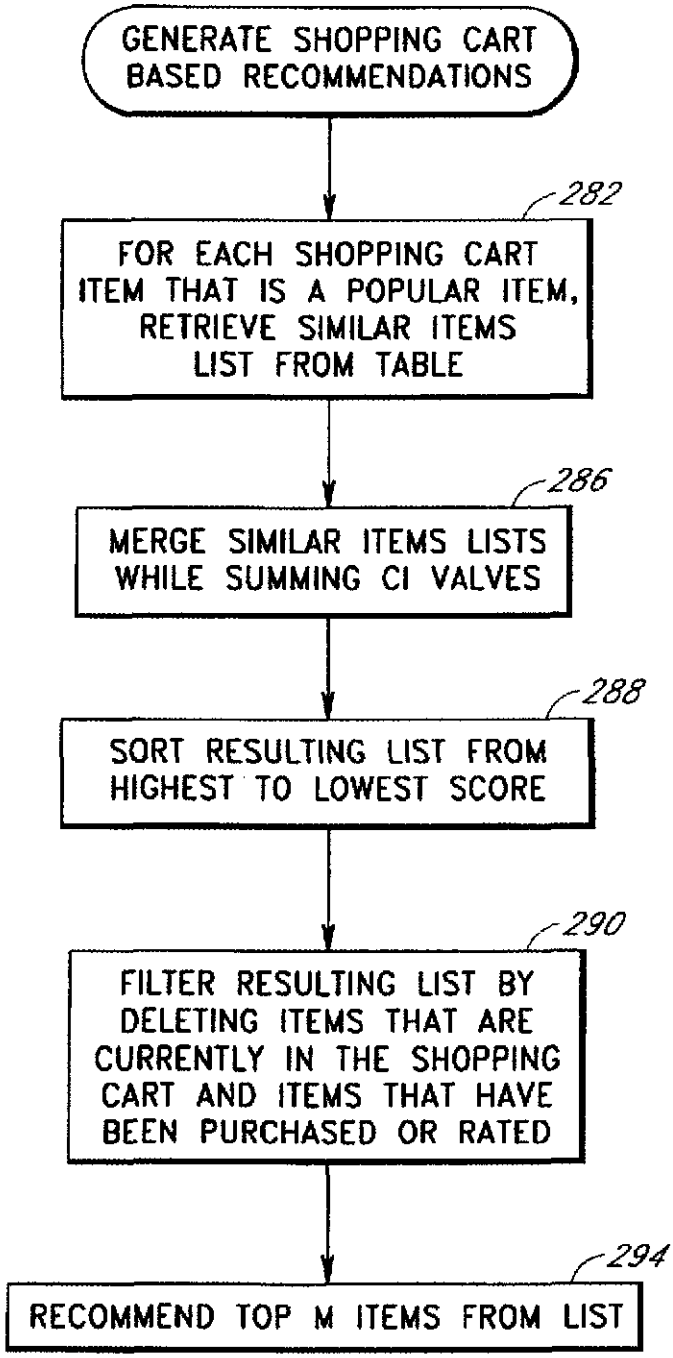

GENERATE SHOPPING CART BASED RECOMMENDATIONS
282
FOR EACH SHOPPING CART ITEM THAT IS A POPULAR ITEM, RETRIEVE SIMILAR ITEMS LIST FROM TABLE
286
MERGE SIMILAR ITEMS LISTS WHILE SUMMING CI VALVES
288
SORT RESULTING LIST FROM HIGHEST TO LOWEST SCORE
290
FILTER RESULTING LIST BY DELETING ITEMS THAT ARE CURRENTLY IN THE SHOPPING CART AND ITEMS THAT HAVE BEEN PURCHASED OR RATED
294
RECOMMEND TOP M ITEMS FROM LIST

FIG. 7

COLLABORATIVE RECOMMENDATIONS USING ITEM-TO-ITEM SIMILARITY MAPPINGS

FIELD OF THE INVENTION

The present invention relates to information filtering and recommendation systems. More specifically, the invention relates to methods for predicting the interests of individual users based on the known interests of a community of users.

BACKGROUND OF THE INVENTION

A recommendation service is a computer-implemented service that recommends items from a database of items. The recommendations are customized to particular users based on information known about the users. One common application for recommendation services involves recommending products to online customers. For example, online merchants commonly provide services for recommending products (books, compact discs, videos, etc.) to customers based on profiles that have been developed for such customers. Recommendation services are also common for recommending Web sites, articles, and other types of informational content to users.

One technique commonly used by recommendation services is known as content-based filtering. Pure content-based systems operate by attempting to identify items which, based on an analysis of item content, are similar to items that are known to be of interest to the user. For example, a content-based Web site recommendation service may operate by parsing the user's favorite Web pages to generate a profile of commonly-occurring terms, and then use this profile to search for other Web pages that include some or all of these terms.

Content-based systems have several significant limitations. For example, content-based methods generally do not provide any mechanism for evaluating the quality or popularity of an item. In addition, content-based methods generally require that the items include some form of content that is amenable to feature extraction algorithms; as a result, content-based systems tend to be poorly suited for recommending movies, music titles, authors, restaurants, and other types of items that have little or no useful, parsable content.

Another common recommendation technique is known as collaborative filtering. In a pure collaborative system, items are recommended to users based on the interests of a community of users, without any analysis of item content. Collaborative systems commonly operate by having the users rate individual items from a list of popular items. Through this process, each user builds a personal profile of ratings data. To generate recommendations for a particular user, the user's profile is initially compared to the profiles of other users to identify one or more "similar users." Items that were rated highly by these similar users (but which have not yet been rated by the user) are then recommended to the user. An important benefit of collaborative filtering is that it overcomes the above-noted deficiencies of content-based filtering.

As with content-based filtering methods, however, existing collaborative filtering techniques have several problems. One problem is that the user is commonly faced with the onerous task of having to rate items in the database to build up a personal ratings profile. This task can be frustrating, particularly if the user is not familiar with many of the items that are presented for rating purposes. Further, because collaborative filtering relies on the existence of other, similar users, collaborative systems tend to be poorly suited for providing recommendations to users that have unusual tastes.

Another problem with collaborative filtering techniques is that an item in the database normally cannot be recommended until the item has been rated. As a result, the operator of a new collaborative recommendation system is commonly faced with a "cold start" problem in which the service cannot be brought online in a useful form until a threshold quantity of ratings data has been collected. In addition, even after the service has been brought online, it may take months or years before a significant quantity of the database items can be recommended.

Another problem with collaborative filtering methods is that the task of comparing user profiles tends to be time consuming —particularly if the number of users is large (e.g., tens or hundreds of thousands). As a result, a tradeoff tends to exist between response time and breadth of analysis. For example, in a recommendation system that generates real-time recommendations in response to requests from users, it may not be feasible to compare the user's ratings profile to those of all other users. A relatively shallow analysis of the available data (leading to poor recommendations) may therefore be performed.

Another problem with both collaborative and content-based systems is that they generally do not reflect the current preferences of the community of users. In the context of a system that recommends products to customers, for example, there is typically no mechanism for favoring items that are currently "hot sellers." In addition, existing systems do not provide a mechanism for recognizing that the user may be searching for a particular type or category of item.

SUMMARY OF THE DISCLOSURE

The present invention addresses these and other problems by providing a computer-implemented service and associated methods for generating personalized recommendations of items based on the collective interests of a community of users. An important benefit of the service is that the recommendations are generated without the need for the user, or any other users, to rate items. Another important benefit is that the recommended items are identified using a previously-generated table or other mapping structure which maps individual items to lists of "similar" items. The item similarities reflected by the table are based at least upon correlations between the interests of users in particular items.

The types of items that can be recommended by the service include, without limitation, books, compact discs ("CDs"), videos, authors, artists, item categories Web sites, and chat groups. The service may be implemented, for example, as part of a Web site, online services network, e-mail notification service, document filtering system, or other type of computer system that explicitly or implicitly recommends items to users. In a preferred embodiment described herein, the service is used to recommend works such as book titles and music titles to users of an online merchant's Web site.

In accordance with one aspect of the invention, the mappings of items to similar items ("item-to-item mappings") are generated periodically, such as once per week, by an off-line process which identifies correlations between known interests of users in particular items. For example, in the embodiment described in detail below, the mappings are generating by periodically analyzing user purchase histories to identify correlations between purchases of items. The similarity between two items is preferably measured by determining the number of users that have an interest in both items relative to the number of users

that have an interest in either item (e.g., items A and B are highly similar because a relatively large portion of the users that bought one of the items also bought the other item). The item-to-item mappings could also incorporate other types of similarities, including content-based similarities extracted by analyzing item descriptions or content.

To generate a set of recommendations for a given user, the service retrieves from the table the similar items lists corresponding to items already known to be of interest to the user, and then appropriately combines these lists to generate a list of recommended items. For example, if there are three items that are known to be of interest to the user (such as three items the user recently purchased), the service may retrieve the similar items lists for these three items from the table and combine these lists. Because the item-to-item mappings are regenerated periodically based on up-to-date sales data, the recommendations tend to reflect the current buying trends of the community.

In accordance with another aspect of the invention, the similar items lists read from the table may be appropriately weighted (prior to being combined) based on indicia of the user's affinity for, or current interest in, the corresponding items of known interest. For example, the similar items list for a book that was purchased in the last week may be weighted more heavily than the similar items list for a book that was purchased four months ago. Weighting a similar items list heavily has the effect of increasing the likelihood that the items in that list will be included in the recommendations that are ultimately presented to the user.

An important aspect of the service is that the relatively computation-intensive task of correlating item interests is performed off-line, and the results of this task (item-to-item mappings) stored in a mapping structure for subsequent look-up. This enables the personal recommendations to be generated rapidly and efficiently (such as in real-time in response to a request by the user), without sacrificing breadth of analysis.

Another feature of the invention involves using the current and/or recent contents of the user's shopping cart as inputs to the recommendation service (or to another type of recommendation service which generates recommendations given a unary listing of items). For example, if the user currently has three items in his or her shopping cart, these three items can be treated as the items of known interest for purposes of generating recommendations, in which case the recommendations may be generated and displayed automatically when the user views the shopping cart contents. Using the current and/or recent shopping cart contents as inputs tends to produce recommendations that are highly correlated to the current short-term interests of the user—even if these short term interest differ significantly from the user's general preferences. For example, if the user is currently searching for books on a particular topic and has added several such books to the shopping cart, this method will more likely produce other books that involve the same or similar topics.

Another feature of the invention involves allowing the user to create multiple shopping carts under a single account (such as shopping carts for different family members), and generating recommendations that are specific to a particular shopping cart. For example, the user can be prompted to select a particular shopping cart (or set of shopping carts), and the recommendations can then be generated based on the items that were purchased from or otherwise placed into the designated shopping cart(s). This feature of the invention allows users to obtain recommendations that correspond to the role or purpose (e.g., work versus pleasure) of a particular shopping cart.

Two specific implementations of the service are disclosed, both of which generate personal recommendations using the same type of table. In the first implementation, the recommendations are based on the items that have recently been rated or purchased by the user. In the second implementation, the recommendations are based on the current shopping cart contents of the user.

BRIEF DESCRIPTION OF THE DRAWINGS

These and other features of the invention will now be described with reference to the drawings summarized below. These drawings and the associated description are provided to illustrate a preferred embodiment of the invention, and not to limit the scope of the invention.

FIG. 1 illustrates a Web site which implements a recommendation service which operates in accordance with the invention, and illustrates the flow of information between components.

FIG. 2 illustrates a sequence of steps that are performed by the recommendation process of FIG. 1 to generate personalized recommendations.

FIG. 3 illustrates a sequence of steps that are performed by the table generation process of FIG. 1 to generate a similar items table, and illustrates temporary data structures generated during the process.

FIG. 4 is a Venn diagram illustrating a hypothetical purchase history profile of three items.

FIG. 5 illustrates one specific implementation of the sequence of steps of FIG. 2.

FIG. 6 illustrates the general form of a Web pages used to present the recommendations of the FIG. 5 process to the user.

FIG. 7 illustrates another specific implementation of the sequence of steps of FIG. 2.

DETAILED DESCRIPTION OF PREFERRED EMBODIMENTS

The various features and methods of the invention will now be described in the context of a recommendation service, including two specific implementations thereof, that is used to recommend book titles, music titles, video titles, and other types of items to individual users of the Amazon.com Web site. As will be recognized to those skilled in the art, the disclosed methods can also be used to recommend other types of items, including non-physical items. By way of example and not limitation, the disclosed methods can also be used to recommend authors, artists, categories or groups of titles, Web sites, chat groups, movies, television shows, downloadable content, restaurants, and other users.

Throughout the description, reference will be made to various implementation-specific details of the recommendation service, the Amazon.com Web site, and other recommendation services of the Web site. These details are provided in order to fully illustrate preferred embodiments of the invention, and not to limit the scope of the invention. The scope of the invention is set forth in the appended claims.

I. Overview of Web Site and Recommendation Services

The Amazon.com Web site includes functionality for allowing users to search, browse, and make purchases from an online catalog of several million book titles, music titles, video titles, and other types of items. Using a shopping cart feature of the site, users can add and remove items to/from a personal shopping cart which is persistent over multiple sessions. (As used herein, a "shopping cart" is a data structure and associated code which keeps track of items that

have been selected by a user for possible purchase.) For example, a user can modify the contents of the shopping cart over a period of time, such as one week, and then proceed to a check out area of the site to purchase the shopping cart contents.

The user can also create multiple shopping carts within a single account. For example, a user can set up separate shopping carts for work and home, or can set up separate shopping carts for each member of the user's family. A preferred shopping cart scheme for allowing users to set up and use multiple shopping carts is disclosed in U.S. application Ser. No. 09/104,942, filed Jun. 25, 1998, titled METHOD AND SYSTEM FOR ELECTRONIC COMMERCE USING MULTIPLE ROLES, the disclosure of which is hereby incorporated by reference.

The site also implements a variety of different recommendation services for recommending book titles, music titles, and/or video titles to users. One such service, known as BookMatcher™, allows users to interactively rate individual books on a scale of 1–5 to create personal item ratings profiles, and applies collaborative filtering techniques to these profiles to generate personal recommendations. The BookMatcher service is described in detail in U.S. application Ser. No. 09/040,171 filed Mar. 17, 1998, the disclosure of which is hereby incorporated by reference. The site may also include associated services that allow users to rate other types of items, such as CDs and videos. As described below, the ratings data collected by the BookMatcher service and similar services is optionally incorporated into the recommendation processes of the present invention.

Another type of service is a recommendation service which operates in accordance with the invention. The service ("Recommendation Service") is preferably used to recommend book titles, music titles and/or videos titles to users, but could also be used in the context of the same Web site to recommend other types of items, including authors, artists, and groups or categories of titles. Briefly, given a unary listing of items that are "known" to be of interest to a user (e.g., a list of items purchased, rated, and/or viewed by the user), the Recommendation Service generates a list of additional items ("recommendations") that are predicted to be of interest to the user. (As used herein, the term "interest" refers generally to a user's liking of or affinity for an item; the term "known" is used to distinguish items for which the user has implicitly or explicitly indicated some level of interest from items predicted by the Recommendation Service to be of interest.)

The recommendations are generated using a table which maps items to lists of "similar" items ("similar items lists"), without the need for users to rate any items (although ratings data may optionally be used). For example, if there are three items that are known to be of interest to a particular user (such as three items the user recently purchased), the service may retrieve the similar items lists for these three items from the table, and appropriately combine these lists (as described below) to generate the recommendations.

In accordance with one aspect of the invention, the mappings of items to similar items ("item-to-item mappings") are generated periodically, such as once per week, from data which reflects the collective interests of the community of users. More specifically, the item-to-item mappings are generated by an off-line process which identifies correlations between known interests of users in particular items. For example, in the embodiment described in detail below, the mappings are generating by analyzing user purchase histories to identify correlations between purchases of particular items (e.g., items A and B are similar because a relatively large portion of the users that purchased item A also bought item B). The item-to-item mappings could also reflect other types of similarities, including content-based similarities extracted by analyzing item descriptions or content.

An important aspect of the Recommendation Service is that the relatively computation-intensive task of correlating item interests is performed off-line, and the results of this task (item-to-item mappings) are stored in a mapping structure for subsequent look-up. This enables the personal recommendations to be generated rapidly and efficiently (such as in real-time in response to a request by the user), without sacrificing breadth of analysis.

In accordance with another aspect of the invention, the similar items lists read from the table are appropriately weighted (prior to being combined) based on indicia of the user's affinity for or current interest in the corresponding items of known interest. For example, in one embodiment described below, if the item of known interest was previously rated by the user (such as through use of the BookMatcher service), the rating is used to weight the corresponding similar items list. Similarly, the similar items list for a book that was purchased in the last week may be weighted more heavily than the similar items list for a book that was purchased four months ago.

Another feature of the invention involves using the current and/or recent contents of the user's shopping cart as inputs to the Recommendation Service. For example, if the user currently has three items in his or her shopping cart, these three items can be treated as the items of known interest for purposes of generating recommendations, in which case the recommendations may be generated and displayed automatically when the user views the shopping cart contents. If the user has multiple shopping carts, the recommendations are preferably generated based on the contents of the shopping cart implicitly or explicitly designated by the user, such as the shopping cart currently being viewed. This method of generating recommendations can also be used within other types of recommendation systems, including content-based systems and systems that do not use item-to-item mappings.

Using the current and/or recent shopping cart contents as inputs tends to produce recommendations that are highly correlated to the current short-term interests of the user—even if these short term interests are not reflected by the user's purchase history. For example, if the user is currently searching for a father's day gift and has selected several books for prospective purchase, this method will have a tendency to identify other books that are well suited for the gift recipient.

Another feature of the invention involves generating recommendations that are specific to a particular shopping cart. This allows a user who has created multiple shopping carts to conveniently obtain recommendations that are specific to the role or purpose to the particular cart. For example, a user who has created a personal shopping cart for buying books for her children can designate this shopping cart to obtain recommendations of children's books. In one embodiment of this feature, the recommendations are generated based solely upon the current contents of the shopping cart selected for display. In another embodiment, the user may designate one or more shopping carts to be used to generate the recommendations, and the service then uses the items that were purchased from these shopping carts as the items of known interest.

As will be recognized by those skilled in the art, the above-described techniques for using shopping cart contents to generate recommendations can also be incorporated into other types of recommendation systems, including pure content-based systems.

FIG. **1** illustrates the basic components of the Amazon.com Web site **30**, including the components used to implement the Recommendation Service. The arrows in FIG. **1** show the general flow of information that is used by the Recommendation Service. As illustrated by FIG. **1**, the Web site **30** includes a Web server application **32** ("Web server") which processes HTTP (Hypertext Transfer Protocol) requests received over the Internet from user computers **34**. The Web server **34** accesses a database **36** of HTML (Hypertext Markup Language) content which includes product information pages and other browsable information about the various products of the catalog. The "items" that are the subject of the Recommendation Service are the titles (regardless of media format such as hardcover or paperback) that are represented within this database **36**.

The Web site **30** also includes a "user profiles" database **38** which stores account-specific information about users of the site. Because a group of individuals can share an account, a given "user" from the perspective of the Web site may include multiple actual users. As illustrated by FIG. **1**, the data stored for each user may include one or more of the following types of information (among other things) that can be used to generate recommendations in accordance with the invention: (a) the user's purchase history, including dates of purchase, (b) the user's item ratings profile (if any), (c) the current contents of the user's personal shopping cart(s), and (d) a listing of items that were recently (e.g., within the last six months) removed from the shopping cart(s) without being purchased ("recent shopping cart contents"). If a given user has multiple shopping carts, the purchase history for that user may include information about the particular shopping cart used to make each purchase; preserving such information allows the Recommendation Service to be configured to generate recommendations that are specific to a particular shopping cart.

As depicted by FIG. **1**, the Web server **32** communicates with various external components **40** of the site. These external components **40** include, for example, a search engine and associated database (not shown) for enabling users to interactively search the catalog for particular items. Also included within the external components **40** are various order processing modules (not shown) for accepting and processing orders, and for updating the purchase histories of the users.

The external components **40** also include a shopping cart process (not shown) which adds and removes items from the users' personal shopping carts based on the actions of the respective users. (The term "process" is used herein to refer generally to one or more code modules that are executed by a computer system to perform a particular task or set of related tasks.) In one embodiment, the shopping cart process periodically "prunes" the personal shopping cart listings of items that are deemed to be dormant, such as items that have not been purchased or viewed by the particular user for a predetermined period of time (e.g. two weeks). The shopping cart process also preferably generates and maintains the user-specific listings of recent shopping cart contents.

The external components **40** also include recommendation service components **44** that are used to implement the site's various recommendation services. Recommendations generated by the recommendation services are returned to the Web server **32**, which incorporates the recommendations into personalized Web pages transmitted to users.

The recommendation service components **44** include a BookMatcher application **50** which implements the above-described BookMatcher service. Users of the BookMatcher service are provided the opportunity to rate individual book titles from a list of popular titles. The book titles are rated according to the following scale:

1=Bad!
2=Not for me
3=OK
4=Liked it
5=Loved it!

Users can also rate book titles during ordinary browsing of the site. As depicted in FIG. **1**, the BookMatcher application **50** records the ratings within the user's items rating profile. For example, if a user of the BookMatcher service gives the book *Into Thin Air* a score of "5," the BookMatcher application **50** would record the item (by ISBN or other identifier) and the score within the user's item ratings profile. The BookMatcher application **50** uses the users' item ratings profiles to generate personal recommendations, which can be requested by the user by selecting an appropriate hyperlink. As described in detail below, the item ratings profiles are also used by an "Instant Recommendations" implementation of the Recommendation Service.

The recommendation services components **44** also include a recommendation process **52**, a similar items table **60**, and an off-line table generation process **66**, which collectively implement the Recommendation Service. As depicted by the arrows in FIG. **1**, the recommendation process **52** generates personal recommendations based on information stored within the similar items table **60**, and based on the items that are known to be of interest ("items of known interest") to the particular user.

In the embodiments described in detail below, the items of known interest are identified based on information stored in the user's profile, such as by selecting all items purchased by the user or all items in the user's shopping cart. In other embodiments of the invention, other types of methods or sources of information could be used to identify the items of known interest. For example, in a service used to recommend Web sites, the items (Web sites) known to be of interest to a user could be identified by parsing a Web server access log and/or by extracting URLs from the "favorite places" list of the user's Web browser. In a service used to recommend restaurants, the items (restaurants) of known interest could be identified by parsing the user's credit card records to identify restaurants that were visited more than once.

The various processes **50**, **52**, **66** of the recommendation services may run, for example, on one or more Unix or NT based workstations or physical servers (not shown) of the Web site **30**. The similar items table **60** is preferably stored as a B-tree data structure to permit efficient look-up, and may be replicated across multiple machines (together with the associated code of the recommendation process **52**) to accommodate heavy loads.

II. Similar Items Table (FIG. **1**)

The general form and content of the similar items table **60** will now be described with reference to FIG. **1**. As this table can take on many alternative forms, the details of the table are intended to illustrate, and not limit, the scope of the invention.

As indicated above, the similar items table **60** maps items to lists of similar items based at least upon the collective interests of the community of users. The similar items table **60** is preferably generated periodically (e.g., once per week)

by the off-line table generation process **66**. The table generation process **66** generates the table **60** from data that reflects the collective interests of the community of users. In the embodiment described in detail herein, the similar items table is generated exclusively from the purchase histories of the community of users (as depicted in FIG. **1**). In other embodiments, the table **60** may additionally or alternatively be generated from other indicia of user-item interests, including indica based on users viewing activities, shopping cart activities, and item rating profiles. For example, the table **60** could be built exclusively from the present and/or recent shopping cart contents of users. The similar items table **60** could also reflect non-collaborative type item similarities, including content-based similarities derived by comparing item contents or descriptions.

Each entry in the similar items table **60** is preferably in the form of a mapping of a popular item **62** to a corresponding list **64** of similar items ("similar items lists"). As used herein, a "popular" item is an item which satisfies some pre-specified popularity criteria. For example, in the embodiment described herein, an item is treated as popular of it has been purchased by more than **30** customers during the life of the Web site. Using this criteria produces a set of popular items (and thus a recommendation service) which grows over time. The similar items list **64** for a given popular item **62** may include other popular items.

In other embodiments involving sales of products, the table **60** may include entries for most or all of the products of the online merchant, rather than just the popular items. In the embodiment described herein, several different types of items (books, CDs, videos, etc.) are reflected within the same table **60**, although separate tables could alternatively be generated for each type of item.

Each similar items list **64** consists of the N (e.g., 20) items which, based on correlations between purchases of items, are deemed to be the most closely related to the respective popular item **62**. Each item in the similar items list **64** is stored together with a commonality index ("CI") value which indicates the relatedness of that item to the popular item **62**, based on sales of the respective items. A relatively high commonality index for a pair of items ITEM A and ITEM B indicates that a relatively large percentage of users who bought ITEM A also bought ITEM B (and vice versa). A relatively low commonality index for ITEM A and ITEM B indicates that a relatively small percentage of the users who bought ITEM A also bought ITEM B (and vice versa). As described below, the similar items lists are generated, for each popular item, by selecting the N other items that have the highest commonality index values. Using this method, ITEM A may be included in ITEM B's similar items list even though ITEM B in not present in ITEM A's similar items list.

In the embodiment depicted by FIG. **1**, the items are represented within the similar items table **60** using product IDs, such as ISBNs or other identifiers. Alternatively, the items could be represented within the table by title ID, where each title ID corresponds to a given "work" regardless of its media format. In either case, different items which correspond to the same work, such as the hardcover and paperback versions of a given book or the VCR cassette and DVD versions of a given video, are preferably treated as a unit for purposes of generating recommendations.

Although the recommendable items in the described system are in the form of book titles, music titles and videos titles, it will be appreciated that the underlying methods and data structures can be used to recommend a wide range of other types of items. For example, in the system depicted by FIG. **1**, the Recommendation Service could also be used to recommend authors, artists, and categorizations or groups of works.

III. General Process for Generating Recommendations (FIG. **2**)

The general sequence of steps that are performed by the recommendation process **52** to generate a set of personal recommendations will now be described with reference to FIG. **2**. This process, and the more specific implementations of the process depicted by FIGS. **5** and **7** (described below), are intended to illustrate, and not limit, the scope of the invention.

The FIG. **2** process is preferably invoked in real-time in response to an online action of the user. For example, in an Instant Recommendations implementation (FIGS. **5** and **6**) of the service, the recommendations are generated and displayed in real-time (based on the user's purchase history and/or item ratings profile) in response to selection by the user of a corresponding hyperlink, such as a hyperlink which reads "Instant Book Recommendations" or "Instant Music Recommendations." In a shopping cart based implementation (FIG. **7**), the recommendations are generated (based on the user's current and/or recent shopping cart contents) in real-time when the user initiates a display of a shopping cart, and are displayed on the same Web page as the shopping cart contents. The Instant Recommendations and shopping cart based embodiments are described separately below under corresponding headings.

Any of a variety of other methods can be used to initiate the recommendations generation process and to display the recommendations to the user. For example, the recommendations can automatically be generated periodically and sent to the user by e-mail, in which case the e-mail listing may contain hyperlinks to the product information pages of the recommended items. Further, the personal recommendations could be generated in advance of any request or action by the user, and cached by the Web site **30** until requested.

As illustrated by FIG. **2**, the first step (step **80**) of the recommendations-generation process involves identifying a set of items that are of known interest to the user. The "knowledge" of the user's interest can be based on explicit indications of interest (e.g., the user rated the item highly) or implicit indications of interest (e.g., the user added the item to a shopping cart). Items that are not "popular items" within the similar items table **60** can optionally be ignored during this step.

In the embodiment depicted in FIG. **1**, the items of known interest are selected from one or more of the following groups: (a) items in the user's purchase history (optionally limited to those items purchased from a particular shopping cart); (b) items in the user's shopping cart (or a particular shopping cart designated by the user), (c) items rated by the user (optionally with a score that exceeds a certain threshold, such as two), and (d) items in the "recent shopping cart contents" list associated with a given user or shopping cart. In other embodiments, the items of known interest may additionally or alternatively be selected based on the viewing activities of the user. For example, the recommendations process **52** could select items that were viewed by the user for an extended period of time and/or viewed more than once. Further, the user could be prompted to select items of interest from a list of popular items.

For each item of known interest, the service retrieves the corresponding similar items list **64** from the similar items table **60** (step **82**), if such a list exists. If no entries exist in the table **60** for any of the items of known interest, the

process **52** may be terminated; alternatively, the process could attempt to identify additional items of interest, such as by accessing other sources of interest information.

In step **84**, the similar items lists **64** are optionally weighted based on information about the user's affinity for the corresponding items of known interest. For example, a similar items list **64** may be weighted heavily if the user gave the corresponding popular item a rating of "5" on a scale or 1–5, or if the user purchased multiple copies of the item. Weighting a similar items list **64** heavily has the effect of increasing the likelihood that the items in that list we be included in the recommendations that are ultimately presented to the user. In one implementation described below, the user is presumed to have a greater affinity for recently purchased items over earlier purchased items.

The similar items lists **64** are preferably weighted by multiplying the commonality index values of the list by a weighting value. The commonality index values as weighted by any applicable weighting value are referred to herein as "scores." In other embodiments, the recommendations may be generated without weighting the similar items lists **64**.

If multiple similar items lists **64** are retrieved in step **82**, the lists are appropriately combined (step **86**), such as by merging the lists while summing the scores of like items. The resulting list is then sorted (step **88**) in order of highest-to-lowest score. In step **90**, the sorted list is filtered to remove unwanted items. The items removed during the filtering process may include, for example, items that have already been purchased or rated by the user, and items that fall outside any product group (such as music or books), product category (such as non-fiction), or content rating (such as PG or adult) designated by the user. The filtering step could alternatively be performed at a different stage of the process, such as during the retrieval of the similar items lists from the table **60**. The result of step **90** is a list ("recommendations list") of other items to be recommended to the user.

In step **92**, one or more additional items are optionally added to the recommendations list. In one embodiment, the items added in step **92** are selected from the set of items (if any) in the user's "recent shopping cart contents" list. As an important benefit of this step, the recommendations include one or more items that the user previously considered purchasing but did not purchase. The items added in step **92** may additionally or alternatively be selected using another recommendations method, such as a content-based method.

Finally, in step **94**, a list of the top M (e.g., 15) items of the recommendations list are returned to the Web server **32** (FIG. **1**). The Web server incorporates this list into one or more Web pages that are returned to the user, with each recommended item being presented as a hypertextual link to the item's product information page. The recommendations may alternatively be conveyed to the user by email, facsimile, or other transmission method. Further, the recommendations could be presented as advertisements for the recommended items.

IV. Generation of Similar Items Table (FIGS. **3** and **4**)

The table-generation process **66** is preferably executed periodically (e.g., once a week) to generate a similar items table **60** that reflects the most recent purchase history data. The recommendation process **52** uses the most recently generated version of the table **60** to generate recommendations.

FIG. **3** illustrates the sequence of steps that are performed by the table generation process **66** to build the similar items table **60**. The general form of temporary data structures that are generated during the process are shown at the right of the drawing. As will be appreciated by those skilled in the art, any of a variety of alternative methods could be used to generate the table **60**.

As depicted by FIG. **3**, the process initially retrieves the purchase histories for all customers (step **100**). Each purchase history is in the general form of the user ID of a customer together with a list of the product IDs (ISBNs, etc.) of the items (books, CDs, videos, etc.) purchased by that customer. In embodiments which support multiple shopping carts within a given account, each shopping cart could be treated as a separate customer for purposes of generating the table. For example, if a given user (or group of users that share an account) purchased items from two different shopping carts within the same account, these purchases could be treated as the purchases of separate users.

The product IDs may be converted to title IDs during this process, or when the table **60** is later used to generate recommendations, so that different versions of an item (e.g., hardcover and paperback) are represented as a single item. This may be accomplished, for example, by using a separate database which maps product IDs to title IDs. To generate a similar items table that strongly reflects the current tastes of the community, the purchase histories retrieved in step **100** can be limited to a specific time period, such as the last six months.

In steps **102** and **104**, the process generates two temporary tables **102A** and **104A**. The first table **102A** maps individual customers to the items they purchased. The second table **104A** maps items to the customers that purchased such items. To avoid the effects of "ballot stuffing," multiple copies of the same item purchased by a single customer are represented with a single table entry. For example, even if a single customer purchased 4000 copies of one book, the customer will be treated as having purchased only a single copy. In addition, items that were sold to an insignificant number (e.g., <15) of customers are preferably omitted or deleted from the tables **102A**, **104B**.

In step **106**, the process identifies the items that constitute "popular" items. This may be accomplished, for example, by selecting from the item-to-customers table **104A** those items that were purchased by more than a threshold number (e.g., 30) of customers. In the context of the Amazon.com Web site, to resulting set of popular items may contain hundreds of thousands or millions of items.

In step **108**, the process counts, for each (popular_item, other_item) pair, the number of customers that are in common. A pseudocode sequence for performing this step is listed in Table 1. The result of step **108** is a table that indicates, for each (popular_item, other_item) pair, the number of customers the two have in common. For example, in the hypothetical table **108A** of FIG. **3**, POPULAR_A and ITEM_B have seventy customers in common, indicating that seventy customers bought both items.

TABLE 1

```
for each popular_item
  for each customer in customers of item
    for each other_item in items of customer
      increment common-customer-count(popular_time, other_item)
```

In step **110**, the process generates the commonality indexes for each (popular_item, other_item) pair in the table **108A**. As indicated above, the commonality index (CI) values are measures of the similarity between two items, with larger CI values indicating greater degrees of similarity. The commonality indexes are preferably generated such that, for a given popular_item, the respective commonality

indexes of the corresponding other_items take into consideration both (a) the number of customers that are common to both items, and (b) the total number of customers of the other_item. A preferred method for generating the commonality index values is set forth in the equation below.

TABLE 1

```
for each popular_item
  for each customer in customers of item
    for each other_item in items of customer
      increment common-customer-count(popular_item, other_item)
```

FIG. **4** illustrates this method in example form. In the FIG. **4** example, item_P (a popular item) has two "other items," item_X and item_Y. Item_P has been purchased by **300** customers, item_X by 300 customers, and item_Y by 30,000 customers. In addition, item_P and item_X have 20 customers in common, and item_P and item_Y have 25 customers in common. Applying the equation above to the values shown in FIG. **4** produces the following results:

$$CI(item\_P, item\_X)=20/\sqrt{(300\times300)}=0.0667$$

$$CI(item\_P, item\_Y)=25/\sqrt{(300\times30{,}000)}=0.0083$$

Thus, even though items P and Y have more customers in common than items P and X, items P and X are treated as being more similar than items P and Y. This result desirably reflects the fact that the percentage of item_X customers that bought item_P (6.7%) is much greater than the percentage of item_Y customers that bought item_P (0.08%).

Because this equation is symmetrical (i.e., CI(item_A, item_B)=CI(item_B, item_A)), it is not necessary to separately calculate the CI value for every location in the table **108**A. In other embodiments, an asymmetrical method may be used to generate the CI values. For example, the CI value for a (popular_item, other_item) pair could be generated as (customers of popular_item and other_item)/(customers of other_item).

Following step **110** of FIG. **3**, each popular item has a respective "other_items" list which includes all of the other_items from the table **108**A and their associated CI values. In step **112**, each other_items list is sorted from highest-to-lowest commonality index. Using the FIG. **4** values as an example, item_X would be positioned closer to the top of the item_B's list than item_Y, since 0.014907>0.001643.

In step **114**, the sorted other_items lists are filtered by deleting all list entries that have fewer than 3 customers in common. For example, in the other_items list for POPULAR_A in table **108**A, ITEM_A would be deleted since POPULAR_A and ITEM_A have only two customers in common. Deleting such entries tends to reduce statistically poor correlations between item sales.

In step **116**, the sorted other_items lists are truncated to length N to generate the similar items lists, and the similar items lists are stored in a B-tree table structure for efficient look-up

As indicated above, any of a variety of other methods for evaluating similarities between items could be incorporated into the table generation process **66**. For example, the table generation process could compare item contents and/or use previously-assigned product categorizations as additional indicators of item similarities. An important benefit of the FIG. **3** method, however, is that the items need not contain any content that is amenable to feature extraction techniques, and need not be pre-assigned to any categories. For example, the method can be used to generate a similar items table given nothing more than the product IDs of a set of products and user purchase histories with respect to these products.

Another important benefit of the Recommendation Service is that the bulk of the processing (the generation of the similar items table **60**) is performed by an off-line process. Once this table has been generated, personalized recommendations can be generated rapidly and efficiently, without sacrificing breadth of analysis.

V. Instant Recommendations Service (FIGS. **5** and **6**)

A specific implementation of the Recommendation Service, referred to herein as the Instant Recommendations service, will now be described with reference to FIGS. **5** and **6**.

As indicated above, the Instant Recommendations service is invoked by the user by selecting a corresponding hyperlink from a Web page. For example, the user may select an "Instant Book Recommendations" or similar hyperlink to obtain a listing of recommended book titles, or may select a "Instant Music Recommendations" or "Instant Video Recommendations" hyperlink to obtain a listing of recommended music or video titles. As described below, the user can also request that the recommendations be limited to a particular item category, such as "non-fiction," "jazz" or "comedies." The Instant Recommendations service generates the recommendations based exclusively on the purchase history and any item ratings profile of the particular user. The service becomes available to the user (i.e., the appropriate hyperlink is presented to the user) once the user has purchased and/or rated a threshold number (e.g. three) of popular items within the corresponding product group. If the user has established multiple shopping carts, the user may also be presented the option of designating a particular shopping cart to be used in generating the recommendations.

FIG. **5** illustrates the sequence of steps that are performed by the Instant Recommendations service to generate personal recommendations. Steps **180**–**194** in FIG. **5** correspond, respectively, to steps **80**–**94** in FIG. **2**. In step **180**, the process **52** identifies all popular items that have been purchased by the user (from a particular shopping cart, if designated) or rated by the user, within the last six months. In step **182**, the process retrieves the similar items lists **64** for these popular items from the similar items table **60**.

In step **184**, the process **52** weights each similar items list based on the duration since the associated popular item was purchased by the user (with recently-purchased items weighted more heavily), or if the popular item was not purchased, the rating given to the popular item by the user. The formula used to generate the weight values to apply to each similar items list is listed in C in Table 2. In this formula, "is_purchased" is a boolean variable which indicates whether the popular item was purchased, "rating" is the rating value (1–5), if any, assigned to the popular item by the user, "order_date" is the date/time (measured in seconds since 1970) the popular item was purchased, "now" is the current date/time (measured in seconds since 1970), and "6 months" is six months in seconds.

TABLE 2

```
1  Weight = ( (is_purchased ? 5:rating) * 2 - 5)*
2  (1 + (max( (is purchased ? order_date:0)-(now-6 months), 0) )
3  / (6 months))
```

In line 1 of the formula, if the popular item was purchased, the value "5" (the maximum possible rating value) is selected; otherwise, the user's rating of the item is selected. The selected value (which may range from 1–5) is

then multiplied by 2, and 5 is subtracted from the result. The value calculated in line 1 thus ranges from a minimum of −3 (if the item was rated a ("1") to a maximum of 5 (if the item was purchased or was rated a "5").

The value calculated in line 1 is multiplied by the value calculated in lines 2 and 3, which can range from a minimum of 1 (if the item was either not purchased or was purchased at least six months ago) to a maximum of 2 (if order_date= now). Thus, the weight can range from a minimum of −6 to a maximum of 10. Weights of zero and below indicate that the user rated the item a "2" or below. Weights higher than 5 indicate that the user actually purchased the item (although a weight of 5 or less is possible even if the item was purchased), with higher values indicating more recent purchases.

The similar items lists **64** are weighted in step **184** by multiplying the CI values of the list by the corresponding weight value. For example, if the weight value for a given popular item is ten, and the similar items list **64** for the popular item is

(productid_A, 0.10), (productid_B, 0.09), (productid_C, 0.08),

the weighted similar items list would be:

(productid_A, 1.0), (productid_BB, 0.9), (productid_C, 0.8),

The numerical values in the weighted similar items lists are referred to as "scores."

In step **186**, the weighted similar items lists are merged (if multiple lists exist) to form a single list. During this step, the scores of like items are summed. For example, if a given other_item appears in three different similar items lists **64**, the three scores (including any negative scores) are summed to produce a composite score.

In step **188**, the resulting list is sorted from highest-to-lowest score. The effect of the sorting operation is to place the most relevant items at the top of the list. In step **190**, the list is filtered by deleting any items that (1) have already been purchased or rated by the user, (2) have a negative score, or (3) do not fall within the designated product group (e.g., books) or category (e.g., "science fiction," or "jazz").

In step **192** one or more items are optionally selected from the recent shopping cart contents list (if such a list exists) for the user, excluding items that have been rated by the user or which fall outside the designated product group or category. The selected items, if any, are inserted at randomly-selected locations within the top M (e.g., 15) positions in the recommendations list. Finally, in step **194**, the top M items from the recommendations list are returned to the Web server **32**, which incorporates these recommendations into one or more Web pages.

The general form of such a Web page is shown in FIG. 6, which lists five recommended items. From this page, the user can select a link associated with one of the recommended items to view the product information page for that item. In addition, the user can select a "more recommendations" button **200** to view additional items from the list of M items. Further, the user can select a "refine your recommendations" link to rate or indicate ownership of the recommended items. Indicating ownership of an item causes the item to be added to the user's purchase history listing.

The user can also select a specific category such as "non-fiction" or "romance" from a drop-down menu **202** to request category-specific recommendations. Designating a specific category causes items in all other categories to be filtered out in step **190** (FIG. 5).

VI. Shopping Cart Based Recommendations (FIG. 7)

Another specific implementation of the Recommendation Service, referred to herein as shopping cart recommendations, will now be described with reference to FIG. 7.

The shopping cart recommendations service is preferably invoked automatically when the user displays the contents of a shopping cart that contains more than a threshold number (e.g., 1) of popular items. The service generates the recommendations based exclusively on the current contents of the shopping cart. As a, result, the recommendations tend to be highly correlated to the user's current shopping interests. In other implementations, the recommendations may also be based on other items that are deemed to be of current interest to the user, such as items in the recent shopping cart contents of the user and/or items recently viewed by the user. Further, other indications of the user's current shopping interests could be incorporated into the process. For example, any search terms typed into the site's search engine during the user's browsing session could be captured and used to perform content-based filtering of the recommended items list

FIG. 7 illustrates the sequence of steps that are performed by the shopping cart recommendations service to generate a set of shopping-cart-based recommendations. In step **282**, the similar items list for each popular item in the shopping cart is retrieved from the similar items table **60**. The similar items list for one or more additional items that are deemed to be of current interest could also be retrieved during this step, such as the list for an item recently deleted from the shopping cart or recently viewed for an extended period of time.

In step **286**, these similar items lists are merged while summing the commonality index (CI) values of the list items. In step **288**, the resulting list is sorted from highest-to-lowest score. In step **290**, the list is filtered to remove any items that exist in the shopping cart or have been purchased or rated by the user. Finally, in step **294**, the top M (e.g., 5) items of the list are returned as recommendations. The recommendations are preferably presented to the user on the same Web page (not shown) as the shopping cart contents.

If the user has defined multiple shopping carts, the recommendations generated by the FIG. 7 process may be based solely on the contents of the shopping cart currently selected for display. As described above, this allows the user to obtain recommendations that correspond to the role or purpose of a particular shopping cart (e.g., work versus home).

The various uses of shopping cart contents to generate recommendations as described above can be applied to other types of recommendation systems, including content-based systems. For example, the current and/or past contents of a shopping cart can be used to generate recommendations in a system in which mappings of items to lists of similar items are generated from a computer-based comparison of item contents. Methods for performing content-based similarity analyses of items are well known in the art, and are therefore not described herein.

Although this invention has been described in terms of certain preferred embodiments, other embodiments that are apparent to those of ordinary skill in the art are also within the scope of this invention. For example, although the embodiments described herein employ item lists, other programming methods for keeping track of and combining sets of similar items can be used. Accordingly, the scope of the present invention is intended to be defined only by reference to the appended claims.

In the claims which follow, reference characters used to denote process steps are provided for convenience of description only, and not to imply a particular order for performing the steps.

What is claimed is:

1. In a multi-user computer system that provides user access to a database of items, a method of recommending items to a user, the method comprising the computer-implemented steps of:

(a) generating a non-user-specific data structure which maps individual items of the database to corresponding sets of similar items in which similarities between items are based at least upon the collective item interests of a community of users;

(b) identifying items that are known to be of interest to the user;

(c) for each of a plurality of the items identified in step (b), accessing the data structure to identify a corresponding set of similar items;

(d) combining the sets of similar items identified in step (c) to generate a combined set of additionally similar items; and

(e) recommending at least some of the similar items of the combined set generated in step (d) to the user;

wherein step (a) is performed in an off-line mode, and steps (b)–(e) are performed substantially in real time in response to an online action by the user.

2. The method of claim 1, wherein step (a) comprises analyzing purchase histories of users to identify correlations between purchases of items.

3. The method of claim 2, wherein step (a) comprises, for each of a plurality of popular items, identifying other items for which a relatively large portion of users that purchased the popular item also purchased the other item.

4. The method of claim 3, wherein step (a) comprises calculating, for each (popular item, other item) pair, a similarity score which reflects a number of users that purchased both the popular item and the other item relative to a number of users that purchased at least one of the popular item and the other item.

5. The method of claim 1, wherein step (a) comprises calculating, for each of a plurality of item pairs (item_A, item_B), a similarity score which reflects a number of users that indicated an interest in both item_A and item_B relative to a number of users that indicated an interest in at least one of item_A and item_B.

6. The method of claim 5, wherein step (a) comprises storing the similarity scores within the data structure, and step (d) comprises using the similarity scores to combine sets of similar items.

7. The method of claim 1, wherein step (a) is repeated periodically, so that item similarities reflected in the data structure reflect current preferences of the community of users.

8. The method of claim 1, wherein the computer system implements online shopping carts for allowing users to select and hold items for possible purchase, and step (b) comprises selecting items that are currently in the user's shopping cart.

9. The method of claim 1, wherein step (b) comprises identifying a plurality of items that were previously purchased by the user.

10. The method of claim 1, wherein step (b) comprises selecting only items that have been placed within a designated shopping cart of a plurality of shopping carts of the user, the method thereby generating recommendations that are specific to a role of the designated shopping cart.

11. The method of claim 1, wherein step (d) comprises weighting at least some of the similar items sets identified in step (c) based on information about the user's affinity for corresponding items of known interest.

12. The method of claim 1, wherein the computer system implements online shopping carts for allowing users to select and hold items for possible purchase, and keeps track of deletions of items from shopping carts, and wherein step (e) further comprises selecting to recommend to the user an item that was previously removed from the user's shopping cart.

13. The method of claim 1, wherein step (a)–(e) are performed without requiring any users to have rated items.

14. The method of claim 1, further comprising filtering out similar items identified in step (c) to remove items already purchased by the user.

15. The method of claim 1, further comprising filtering out similar items identified in step (c) to remove items that do not fall within an item category designated by the user.

16. In a multi-user computer system that provides access to a database of items, a system for recommending items to users, comprising:

a first process which determines similarities between items by at least analyzing historical data that reflects item interests of a community of users, the first process generating a non-user-specific data structure which maps items to sets of similar items; and

a second process which generates personal recommendations for a user by accessing the data structure to identify similar items sets that correspond to items known to be of interest to the user, and by combining the identified similar items sets to generate a list of recommended items;

wherein the first process generates the data structure in an off-line mode, and the second process generates and displays the personal recommendations substantially in real time in response to online actions of users.

17. The system of claim 16, wherein the first process determines a similarity between a pair of items, item_A and item_B, by at least calculating a similarity score which reflects a number of users that are interested in both item_A and item_B relative to a number of users that are interested in item_A or item_B.

18. The system of claim 17, wherein the first process determines a similarity between a pair of items, item_A and item_B, by at least calculating a score value which reflects a number of users that purchased both item_A and item_B relative to a number of users that purchased item_A or item_B.

19. The system of claim 16, wherein the first process generates and stores within the data structure similarity scores that indicate degrees of similarity between items, and the second process uses the similarity scores to combine sets of similar items.

20. The system of claim 16, wherein the first process is an off-line process which executes separately from the second process, and the second process generates recommendations substantially in real-time in response to requests from users.

21. The system of claim 16, wherein the first process is executed periodically to generate a new data structure, so that item similarities reflected in the data structure reflect current preferences of the community of users.

22. The system of claim 16, wherein the computer system implements online shopping carts for allowing users to select and hold items for possible purchase, and wherein the second process generates recommendations for the user based on items that are currently in the user's shopping cart.

23. The system of claim 16, wherein the second process weights at least some of the identified similar items sets based on information about the user's affinity for corresponding items of known interest.

24. The system of claim 16, wherein the computer system allows a user to create multiple shopping carts within a single account, and the second process generates shopping cart specific recommendations to allow a user with multiple shopping carts to obtain recommendations specific to a role of a particular shopping cart.

25. The system of claim 16, wherein the system generates personal recommendations without requiring users to rate items.

26. The system of claim 16, wherein the second process filters out items already purchased by the user from the similar items sets identified from the data structure.

27. The system of claim 16, wherein the second process filters out items from the similar items sets identified from the data structure based on item categories specified by users.

28. In a system for generating personalized recommendations of items from a database of items that are accessed by a community of users, a method of recommending items to users, the method comprising:

in an off-line mode, for each of a plurality of first items:

(a) for each of a plurality of other items of the database, generating a respective score which indicates a degree of similarity between the first item and the other item such that the score is based on at least (i) a number of users that are interested in both the first item and the other item, and (ii) a number of users that are interested in the other item;

(b) sorting the plurality of other items according to the score values generated in step (a);

(c) truncating a list of items which results from step (b); and

(d) storing the truncated list generated in step (c) ("similar items list") together with corresponding scores generated in step (a) in a non-user-specific data structure for subsequent look-up;

subsequently, in response to an action performed by a user, performing the following steps substantially in real time:

(e) for each of a plurality of items that are known to be of interest to the user, accessing the data structure to identify a corresponding similar items list;

(f) combining the similar items lists identified in (e) to generate a combined list of similar items, wherein combining the similar items lists comprises combining scores of like items; and

(g) recommending at least some of the items from the combined list generated in (f) to the user.

29. The method of claim 28, wherein step (a) comprises calculating a score which is further based on the number of users that are interested in the first item.

30. The method as in claim 1, wherein the items are products that are available for online purchase.

31. The system as in claim 16, wherein the items are products that are available for online purchase.

32. The system as in claim 28, wherein the items are products that are available for online purchase.

33. A method of generating instant product recommendations for online users, comprising:

in an off-line mode, generating a data structure which maps each of a plurality of products directly to a corresponding set of similar products in which product similarities are indicated by similarity scores stored within the non-user-specific data structure; and

subsequently, in response to an action by an online user, immediately generating and displaying personal product recommendations for the user by at least (a) accessing the data structure to look up a respective set of similar products and associated similarity scores for each of multiple products known to be of interest to the user, (b) combining the sets of similar products identified in (a) into a ranked set in which rankings are based at least in-part on the similarity scores, and (c) selecting at least some of the products in the ranked set to display to the online user.

34. The method as in claim 33, wherein generating the data structure comprises using customer purchase histories to predict similarities between products.

35. The method as in claim 34, wherein generating the data structure further comprises generating, for each of multiple pairs of products, a similarity score which is based at least on a number of users that purchased both products in the pair.

36. The method as in claim 35, wherein generating the data structure further comprises using the purchase histories to determine popularity levels of products, and using said popularity levels to select the plurality of products for which to identify and store corresponding sets of similar products.

37. The method as in claim 33, wherein the action by the online user is a request to view a personal shopping cart, and the method comprises looking up from the data structure respective similar products sets for each of multiple products represented within the shopping cart.

38. The method as in claim 33, wherein the method comprises inhibiting selection in (c) of products already purchased by the user.

39. The method as in claim 33, wherein the data structure is a B-tree.

40. The method as in claim 33, further comprising replicating the data structure across multiple machines to accommodate heavy loads.

41. The method as in claim 1, wherein (d) comprises generating a ranked set of similar items in which a similar item's ranking reflects whether that similar item appears within more than one of the sets identified in step (c).

42. The system of claim 16, wherein the list of recommended items is a ranked list in which an item's ranking reflects whether that item appears within more than one of said similar items sets.

43. A computer-implemented method of recommending products to users, comprising:

generating, for each of a plurality of pairs of products, a respective score indicating a degree to which the products of the pair are deemed related to one another, wherein the score reflects a frequency with which the products of the pair co-occur within purchase histories of users;

storing the scores in a non-user-specific data structure that maps products to sets of related products; and

using the data structure and the scores to provide personalized product recommendations to each of multiple users.

44. The method as in claim 43, wherein using the data structure and scores to provide personalized product recommendations comprises:

identifying multiple products that are of interest to a user;

for each of the multiple products, accessing the data structure to identify a set of related products, to thereby identify multiple sets of related products;

combining the multiple sets of related products to generate a ranked set of related products in which product rankings reflect corresponding scores within the data structure; and

recommending products to the user from the ranked set.

45. The method as in claim **44**, wherein combining the multiple sets comprises combining scores of like products, so that a product's ranking reflects whether or not that product appears within more than one of the multiple sets.

46. The method as in claim **44**, wherein identifying multiple products that are of interest to the user comprises identifying products that are currently in a shopping cart of the user.

47. The method as in claim **46**, wherein recommending products to the user from the ranked set comprises displaying recommended products within a web page that displays current contents of the shopping cart.

48. The method as in claim **43**, wherein using the data structure and scores to provide personalized product recommendations comprises generating and displaying personal recommendations substantially in real time in response to online actions of users.

49. The method as in claim **43**, wherein the scores are generated and stored within the data structure in an off-line processing mode.

* * * * *

UNITED STATES PATENT AND TRADEMARK OFFICE

CERTIFICATE OF CORRECTION

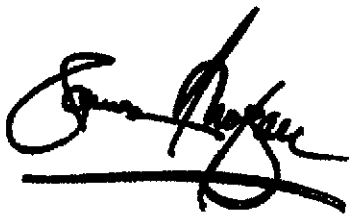
PATENT NO. : 6,266,649 B1
DATED : July 24, 2001
INVENTOR(S) : Linden et al.

It is certified that error appears in the above-identified patent and that said Letters Patent is hereby corrected as shown below:

Column 19,
Line 63, insert -- non-user-specific -- before "data structure".
Line 67, delete "non-user-specific".

Signed and Sealed this

Third Day of December, 2002



JAMES E. ROGAN
Director of the United States Patent and Trademark Office

UNITED STATES PATENT AND TRADEMARK OFFICE

CERTIFICATE OF CORRECTION

PATENT NO. : 6,266,649 B1
DATED : July 24, 2001
INVENTOR(S) : Linden et al.

It is certified that error appears in the above-identified patent and that said Letters Patent is hereby corrected as shown below:

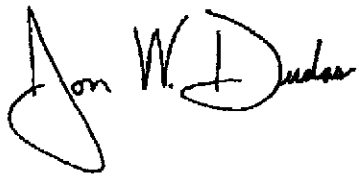
Column 13,
Table 1, replace with the following equation:

$$CI(item\_A,\ item\_B) = \frac{customers\ of\ item\_A\ and\ item\_B}{\sqrt{(customers\ of\ item\_A)\ x\ (customers\ of\ item\_B)}}$$

Column 17,
Line 21, delete "additionally".

Signed and Sealed this

Twenty-seventh Day of April, 2004



JON W. DUDAS
Acting Director of the United States Patent and Trademark Office

EXHIBIT 4

US006317722B1

(12) **United States Patent**
Jacobi et al.

(10) **Patent No.: US 6,317,722 B1**
(45) **Date of Patent: *Nov. 13, 2001**

(54) **USE OF ELECTRONIC SHOPPING CARTS TO GENERATE PERSONAL RECOMMENDATIONS**

(75) Inventors: **Jennifer A. Jacobi**; **Eric A. Benson**; **Gregory D. Linden**, all of Seattle, WA (US)

(73) Assignee: **Amazon.com, Inc.**, Seattle, WA (US)

(*) Notice: This patent issued on a continued prosecution application filed under 37 CFR 1.53(d), and is subject to the twenty year patent term provisions of 35 U.S.C. 154(a)(2).

Subject to any disclaimer, the term of this patent is extended or adjusted under 35 U.S.C. 154(b) by 0 days.

(21) Appl. No.: **09/156,237**

(22) Filed: **Sep. 18, 1998**

(51) **Int. Cl.**[7] **G06F 17/60**; G06F 17/00; G06F 15/173; H04K 1/00; H04H 1/00

(52) **U.S. Cl.** **705/14**; 233/383; 380/24; 455/3.1; 455/480; 705/14; 705/27; 707/102

(58) **Field of Search** 235/383; 380/24; 455/5.1, 480; 705/14, 27; 707/3, 102; 709/227

(56) **References Cited**

U.S. PATENT DOCUMENTS

4,870,579	*	9/1989	Hey	364/419
4,996,642	*	2/1991	Hey	364/419
5,235,509	*	8/1993	Mueller et al.	364/405
5,459,306	*	10/1995	Stein et al.	235/383
5,583,763	*	12/1996	Atcheson et al.	364/551.01
5,745,681	*	4/1998	Levine et al.	395/200.3
5,749,081	*	5/1998	Whiteis	707/102
5,774,670	*	6/1998	Montulli	395/200.57
5,790,935	*	8/1998	Payton	455/5.1
5,905,973	*	5/1999	Yonezawa et al.	705/27
5,909,492	*	6/1999	Payne et al.	380/24

FOREIGN PATENT DOCUMENTS

0751471 A	*	1/1997	(EP)	G06F/17/60
0 827 063 A	*	3/1998	(EP)	G06F/3/00
0265083 A	*	4/1988	(EP)	G09F/27/00
2336925	*	3/1999	(GB)	G06F/17/00

OTHER PUBLICATIONS

"COSMOCOM", Computer Telephony, p. 124, Jul. 1998.*
Brier, S.E., "Smart Devices Peep Into Your Grocery Cart", New York Times Co., Section G, p. 3, col.3, Circuits, Jul. 1998.*
Nash, E.L., "Direct Marketing; Strategy, Planning, Execution", 3rd Ed., McGraw–Hill, Inc., pp. 165, & 365–6, 1994.*
"iCat Electronic Commerce Suite Takes 'Best of Show' Award At WebINNOVATION 97", PR Newswire, Jun. 1997.*
"iCat Corporation: iCat's Commerce Suite Makes Setting Up Shop on Net Even Easier Than High Street", M2 Presswire, Feb. 1997.*

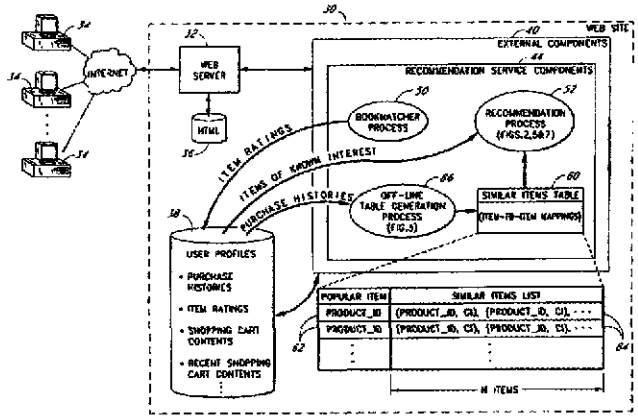
(List continued on next page.)

Primary Examiner—Tariq R. Hafiz
Assistant Examiner—J Harle
(74) *Attorney, Agent, or Firm*—Knobbe, Martens, Olson & Bear, LLP

(57) **ABSTRACT**

A computer-implemented service recommends products or other items to a user based on a set of items known to be of interest to the user, such as a set of items currently in the user's electronic shopping cart. In one embodiment, the service identifies items that are currently in the user's shopping cart, and uses these items to generate a list of additional items that are predicted to be of interest to the user, wherein an additional item is selected to include in the list based in-part upon whether that item is related to more than one of the items in the user's shopping cart. The item relationships are preferably determined by an off-line process that analyzes user purchase histories to identify correlations between item purchases. The additional items are preferably displayed to the user when the user views the contents of the shopping cart.

42 Claims, 7 Drawing Sheets



OTHER PUBLICATIONS

Dragan et al., "Advice From the Web", PC Magazine, v.16, n.15, p. 133, Sep. 1997.*

"Able Solutions Announces Able Commerce 2.6", PR Newswire, Sep. 1998.*

"Internet World—IBM To Expand E-Comm Features", Newsbytes News Network, Dec. 1996.*

McMains, A., "Weiss, Whitten, Staliano's", ADWEEK Eastern Edition, v.39, n.24, p. 82, Jun. 1998.*

"Cdnow Rated Top Music Site by eMarketer, the Authority on Business Online", PR Newswire, Sep. 1998.*

Joaquin Delgado, "Intelligent Collaborative Information Retrieval".*

Joaquin Delgado, "Content-based Collaborative Information Filtering".*

Marko Balabanovic and Yoav Shoham, "Content-Based, Collaborative Recommendation," Communications of the ACM, v 40n3, pp. 66-72, Mar. 1997.*

Upendra Shardanand and Pattie Maes with MIT Media-Lab, Social Information Filtering: Algorithms for Automating "Word of Mouth", 8 pgs. (undated).

Combining Social Networks and Collaborative Filtering, Communications of the ACM, Mar. 1997/vol. 40, No. 3, pp. 63-65.

Pointing the Way: Active Collaborative Filtering, CHI '95 Proceedings Papers, 11 pgs.

Bradley N. Miller, John T. Riedl, Joseph A. Konstan with Department of Computer Science, University of Minnesota, Experiences with GroupLens: Making Usenet Useful Again, 13 pgs.

A System for Sharing Recommendations, Communications of the ACM, Mar. 1997/vol. 40, No. 3, pp. 59-62.

Recommender Systems for Evaluating Computer Messages, Communications of the ACM, Mar. 1997/vol. 40, No. 3, pp. 88 and 89.

Content-Based, Collaborative Recommendation, Communciations of the ACM, Mar. 1997/vol. 40, No. 3, pp. 66-72.

Applying Collaborative Filtering to Usenet News, Communications of the ACM, Mar. 1997/vol. 40, No. 3, pp 77-87.

Personalized Navigation for the Web, Communications of the ACM, Mar. 1997/vol. 40, No. 3, pp. 73-76.

GroupLens: An Open Architecture for Collaborative Filtering of Netnews, 18 pgs.

Net Perceptions, Inc., White Paper, Building Customer Loyalty and High-Yield Relationships Through GroupLens Collaborative Filtering, 9 pgs., Nov. 22, 1996.

Christos Faloutsos and Douglas Oard with University of Maryland, A Survey of Information Retrieval and Filtering Methods, 22 pgs. (undated).

* cited by examiner

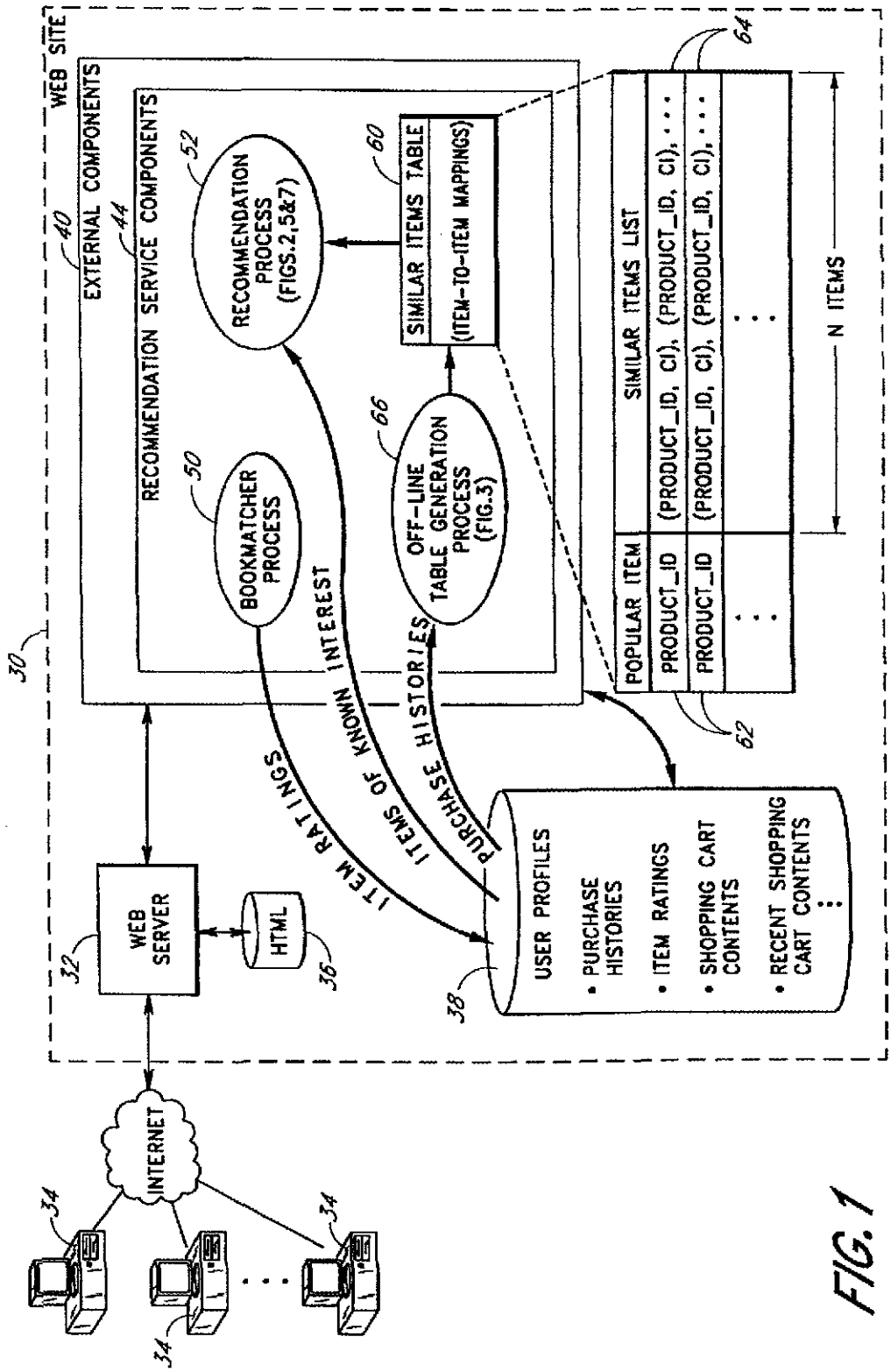

WEB SITE
30
EXTERNAL COMPONENTS
40
RECOMMENDATION SERVICE COMPONENTS
44
RECOMMENDATION PROCESS (FIGS.2,5&7)
52
BOOKMATCHER PROCESS
50
OFF-LINE TABLE GENERATION PROCESS (FIG.3)
66
SIMILAR ITEMS TABLE
(ITEM-TO-ITEM MAPPINGS)
60
POPULAR ITEM
SIMILAR ITEMS LIST
PRODUCT_ID
(PRODUCT_ID, CI), (PRODUCT_ID, CI), . . .
64
N ITEMS
62
ITEM RATINGS
ITEMS OF KNOWN INTEREST
PURCHASE HISTORIES
WEB SERVER
32
HTML
36
USER PROFILES
• PURCHASE HISTORIES
• ITEM RATINGS
• SHOPPING CART CONTENTS
• RECENT SHOPPING CART CONTENTS
38
INTERNET
34

FIG. 1

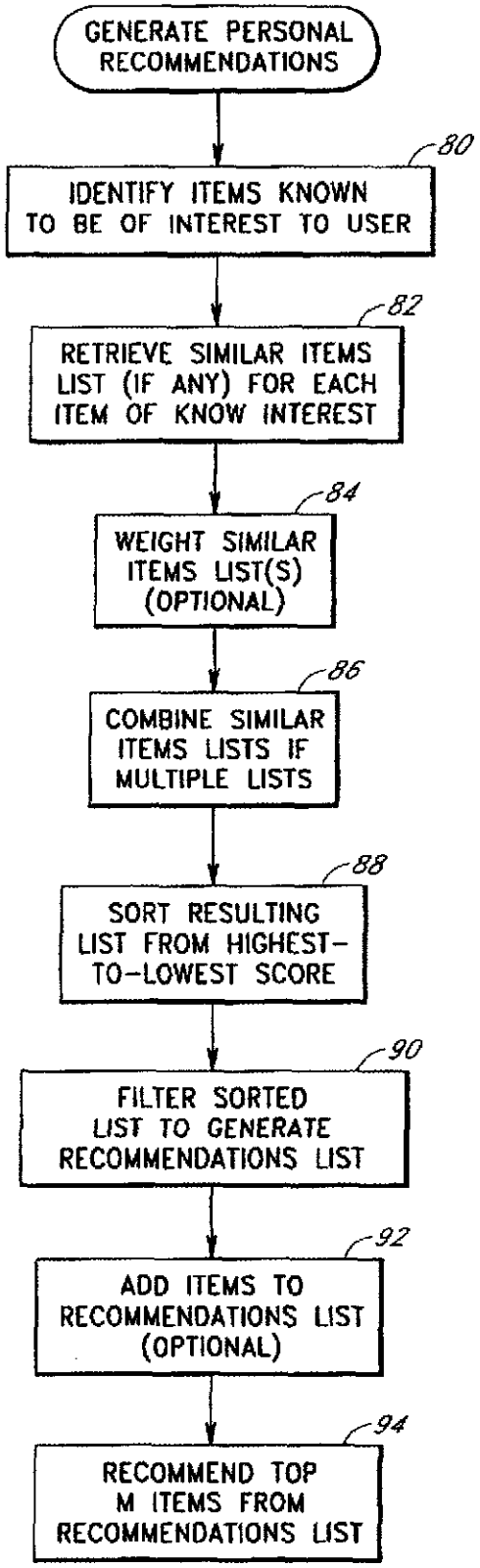

GENERATE PERSONAL RECOMMENDATIONS
80
IDENTIFY ITEMS KNOWN TO BE OF INTEREST TO USER
82
RETRIEVE SIMILAR ITEMS LIST (IF ANY) FOR EACH ITEM OF KNOW INTEREST
84
WEIGHT SIMILAR ITEMS LIST(S) (OPTIONAL)
86
COMBINE SIMILAR ITEMS LISTS IF MULTIPLE LISTS
88
SORT RESULTING LIST FROM HIGHEST-TO-LOWEST SCORE
90
FILTER SORTED LIST TO GENERATE RECOMMENDATIONS LIST
92
ADD ITEMS TO RECOMMENDATIONS LIST (OPTIONAL)
94
RECOMMEND TOP M ITEMS FROM RECOMMENDATIONS LIST

FIG. 2

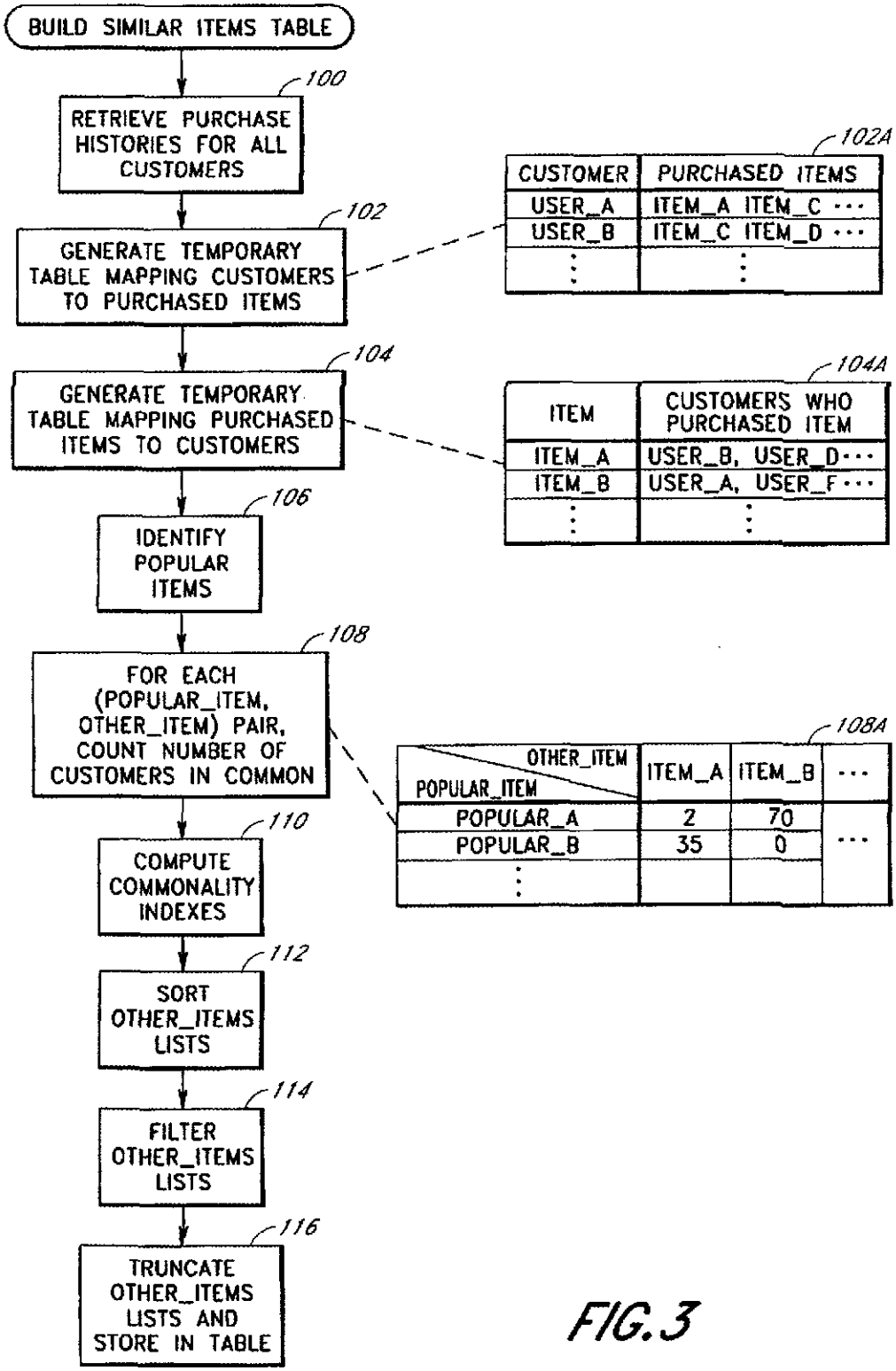

BUILD SIMILAR ITEMS TABLE
100
RETRIEVE PURCHASE HISTORIES FOR ALL CUSTOMERS
102
GENERATE TEMPORARY TABLE MAPPING CUSTOMERS TO PURCHASED ITEMS
102A
CUSTOMER
PURCHASED ITEMS
USER_A
ITEM_A ITEM_C ···
USER_B
ITEM_C ITEM_D ···
104
GENERATE TEMPORARY TABLE MAPPING PURCHASED ITEMS TO CUSTOMERS
104A
ITEM
CUSTOMERS WHO PURCHASED ITEM
ITEM_A
USER_B, USER_D···
ITEM_B
USER_A, USER_F···
106
IDENTIFY POPULAR ITEMS
108
FOR EACH (POPULAR_ITEM, OTHER_ITEM) PAIR, COUNT NUMBER OF CUSTOMERS IN COMMON
108A
OTHER_ITEM
POPULAR_ITEM
ITEM_A
ITEM_B
···
POPULAR_A
2
70
POPULAR_B
35
0
110
COMPUTE COMMONALITY INDEXES
112
SORT OTHER_ITEMS LISTS
114
FILTER OTHER_ITEMS LISTS
116
TRUNCATE OTHER_ITEMS LISTS AND STORE IN TABLE

FIG. 3

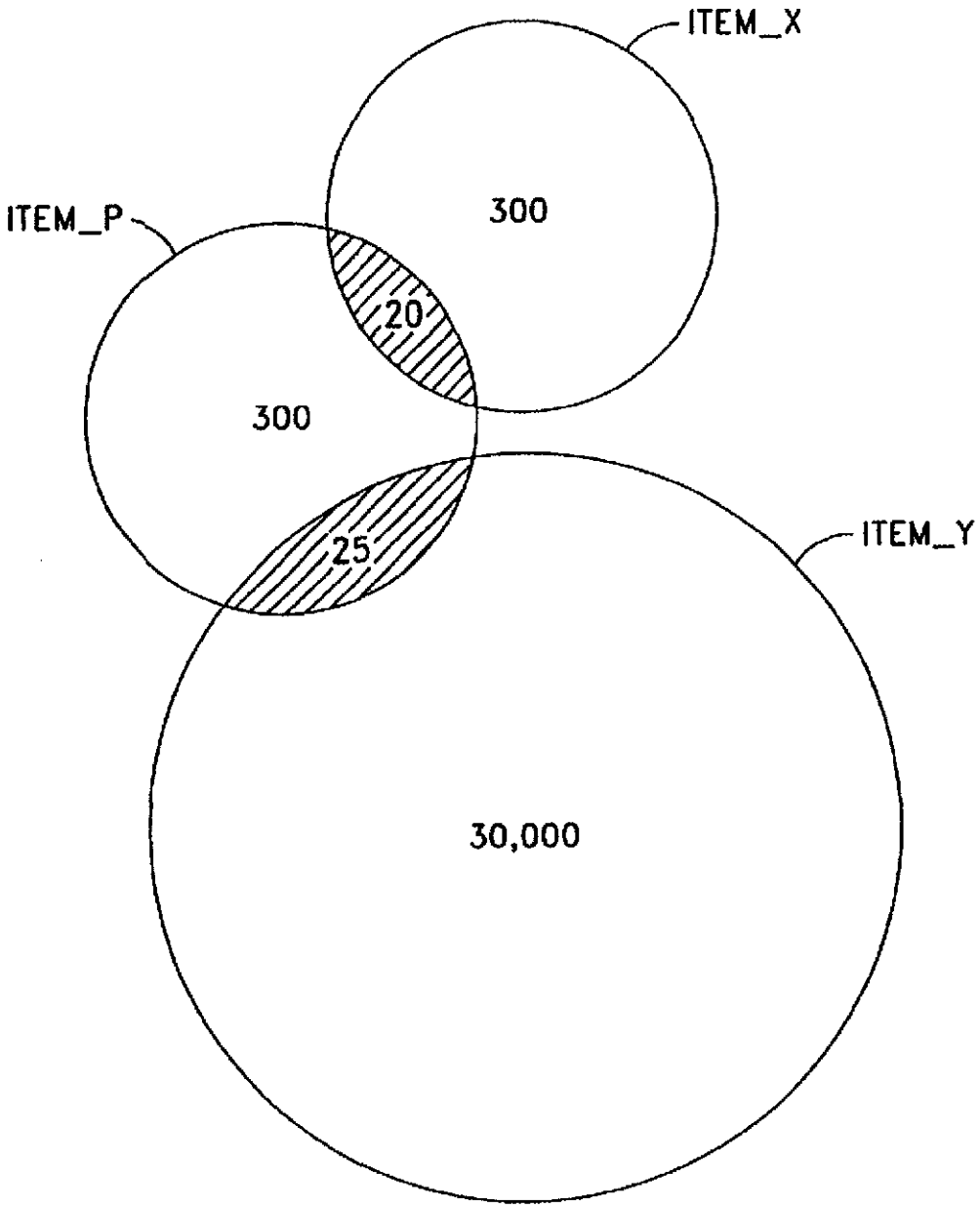


FIG. 4

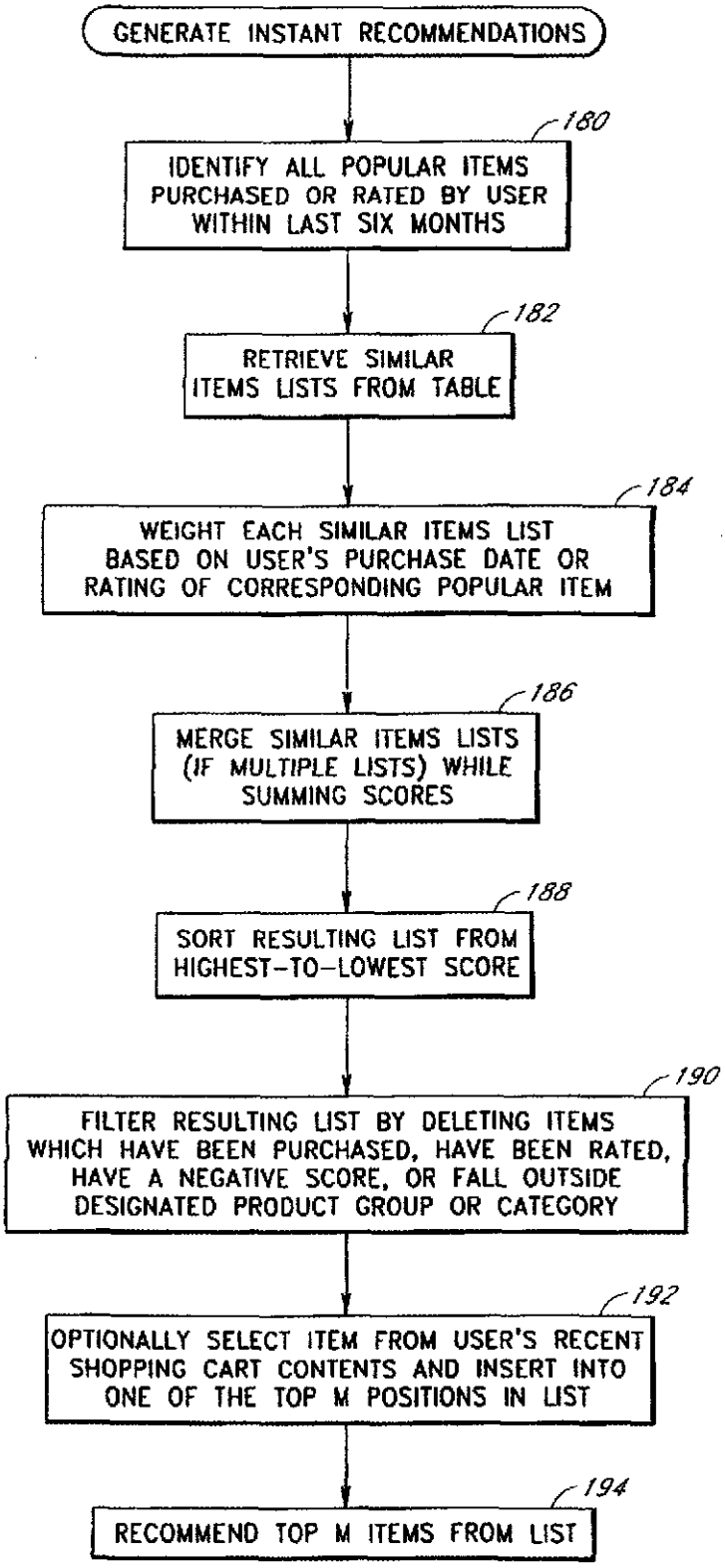

GENERATE INSTANT RECOMMENDATIONS
180
IDENTIFY ALL POPULAR ITEMS PURCHASED OR RATED BY USER WITHIN LAST SIX MONTHS
182
RETRIEVE SIMILAR ITEMS LISTS FROM TABLE
184
WEIGHT EACH SIMILAR ITEMS LIST BASED ON USER'S PURCHASE DATE OR RATING OF CORRESPONDING POPULAR ITEM
186
MERGE SIMILAR ITEMS LISTS (IF MULTIPLE LISTS) WHILE SUMMING SCORES
188
SORT RESULTING LIST FROM HIGHEST-TO-LOWEST SCORE
190
FILTER RESULTING LIST BY DELETING ITEMS WHICH HAVE BEEN PURCHASED, HAVE BEEN RATED, HAVE A NEGATIVE SCORE, OR FALL OUTSIDE DESIGNATED PRODUCT GROUP OR CATEGORY
192
OPTIONALLY SELECT ITEM FROM USER'S RECENT SHOPPING CART CONTENTS AND INSERT INTO ONE OF THE TOP M POSITIONS IN LIST
194
RECOMMEND TOP M ITEMS FROM LIST

FIG. 5

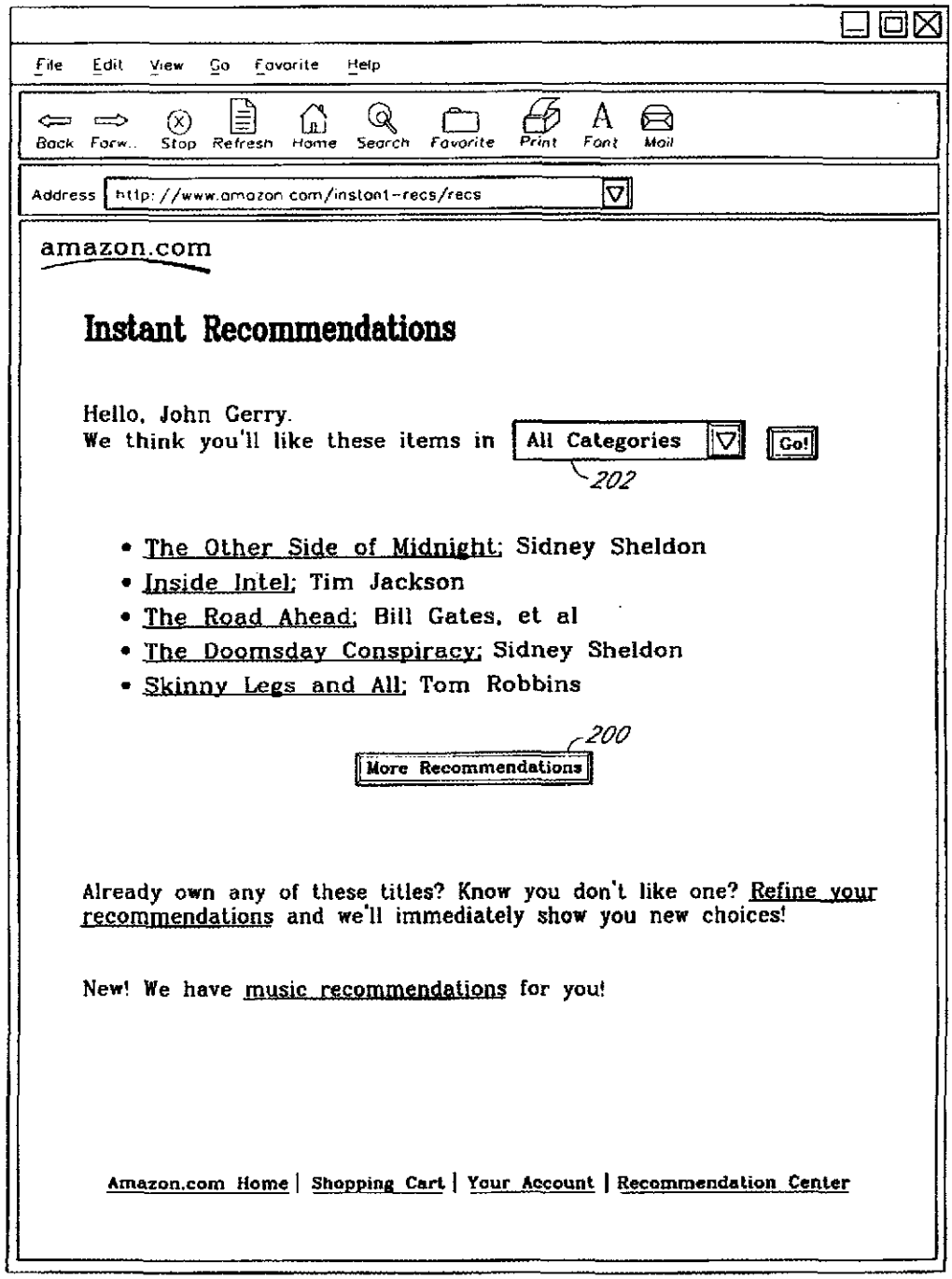

File Edit View Go Favorite Help
Back Forw.. Stop Refresh Home Search Favorite Print Font Mail
Address http://www.amazon.com/instant-recs/recs
amazon.com
Instant Recommendations
Hello, John Gerry.
We think you'll like these items in All Categories Go!
202
The Other Side of Midnight; Sidney Sheldon
Inside Intel; Tim Jackson
The Road Ahead; Bill Gates, et al
The Doomsday Conspiracy; Sidney Sheldon
Skinny Legs and All; Tom Robbins
200
More Recommendations
Already own any of these titles? Know you don't like one? Refine your recommendations and we'll immediately show you new choices!
New! We have music recommendations for you!
Amazon.com Home | Shopping Cart | Your Account | Recommendation Center

FIG. 6

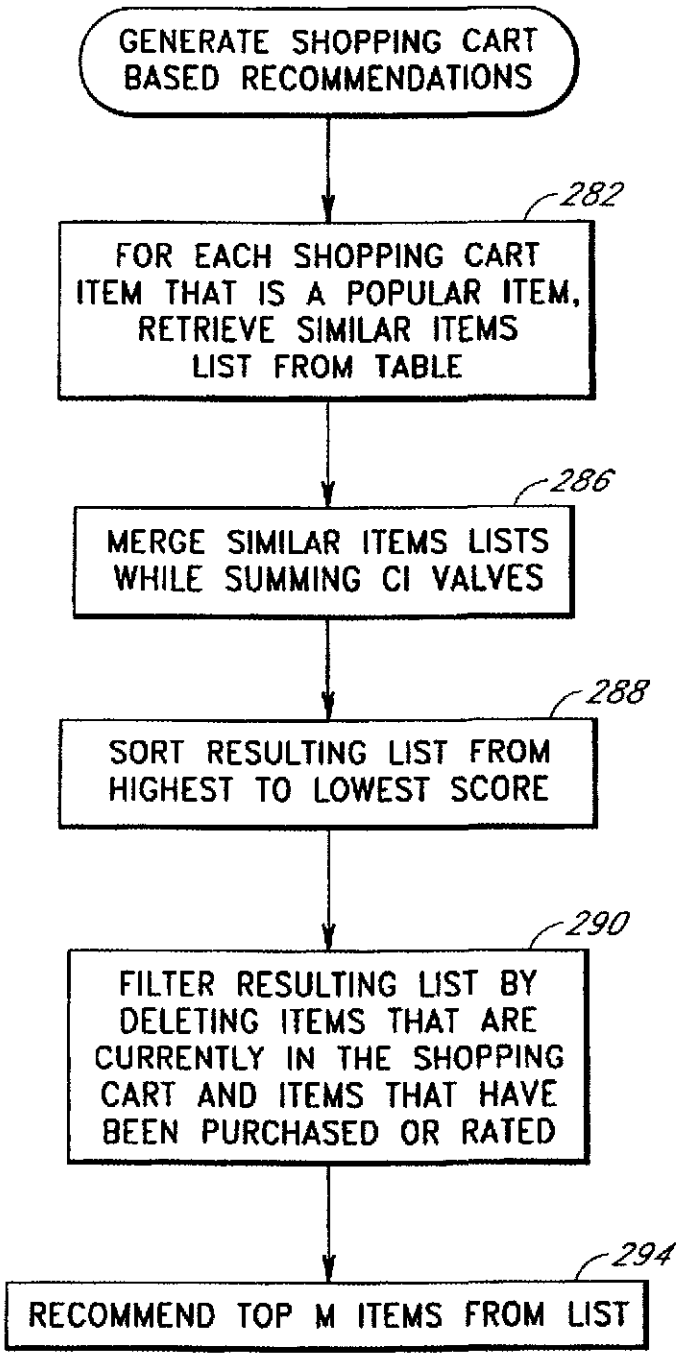

GENERATE SHOPPING CART BASED RECOMMENDATIONS
282
FOR EACH SHOPPING CART ITEM THAT IS A POPULAR ITEM, RETRIEVE SIMILAR ITEMS LIST FROM TABLE
286
MERGE SIMILAR ITEMS LISTS WHILE SUMMING CI VALVES
288
SORT RESULTING LIST FROM HIGHEST TO LOWEST SCORE
290
FILTER RESULTING LIST BY DELETING ITEMS THAT ARE CURRENTLY IN THE SHOPPING CART AND ITEMS THAT HAVE BEEN PURCHASED OR RATED
294
RECOMMEND TOP M ITEMS FROM LIST

FIG. 7

USE OF ELECTRONIC SHOPPING CARTS TO GENERATE PERSONAL RECOMMENDATIONS

FIELD OF THE INVENTION

The present invention relates to information filtering and recommendation systems. More specifically, the invention relates to methods for recommending products or other items to individual users of an electronic commerce system.

BACKGROUND OF THE INVENTION

A recommendation service is a computer-implemented service that recommends items from a database of items. The recommendations are customized to particular users based on information known about the users. One common application for recommendation services involves recommending products to online customers. For example, online merchants commonly provide services for recommending products (books, compact discs, videos, etc.) to customers based on profiles that have been developed for such customers. Recommendation services are also common for recommending Web sites, articles, and other types of informational content to users.

One technique commonly used by recommendation services is known as content-based filtering. Pure content-based systems operate by attempting to identify items which, based on an analysis of item content, are similar to items that are known to be of interest to the user. For example, a content-based Web site recommendation service may operate by parsing the user's favorite Web pages to generate a profile of commonly-occurring terms, and then use this profile to search for other Web pages that include some or all of these terms.

Content-based systems have several significant limitations. For example, content-based methods generally do not provide any mechanism for evaluating the quality or popularity of an item. In addition, content-based methods generally require that the items include some form of content that is amenable to feature extraction algorithms; as a result, content-based systems tend to be poorly suited for recommending movies, music titles, authors, restaurants, and other types of items that have little or no useful, parsable content.

Another common recommendation technique is known as collaborative filtering. In a pure collaborative system, items are recommended to users based on the interests of a community of users, without any analysis of item content. Collaborative systems commonly operate by having the users rate individual items from a list of popular items. Through this process, each user builds a personal profile of ratings data. To generate recommendations for a particular user, the user's profile is initially compared to the profiles of other users to identify one or more "similar users." Items that were rated highly by these similar users (but which have not yet been rated by the user) are then recommended to the user. An important benefit of collaborative filtering is that it overcomes the above-noted deficiencies of content-based filtering.

As with content-based filtering methods, however, existing collaborative filtering techniques have several problems. One problem is that the user is commonly faced with the onerous task of having to rate items in the database to build up a personal ratings profile. This task can be frustrating, particularly if the user is not familiar with many of the items that are presented for rating purposes. Further, because collaborative filtering relies on the existence of other, similar users, collaborative systems tend to be poorly suited for providing recommendations to users that have unusual tastes.

Another problem with collaborative filtering techniques is that an item in the database normally cannot be recommended until the item has been rated. As a result, the operator of a new collaborative recommendation system is commonly faced with a "cold start" problem in which the service cannot be brought online in a useful form until a threshold quantity of ratings data has been collected. In addition, even after the service has been brought online, it may take months or years before a significant quantity of the database items can be recommended.

Another problem with collaborative filtering methods is that the task of comparing user profiles tends to be time consuming—particularly if the number of users is large (e.g., tens or hundreds of thousands). As a result, a tradeoff tends to exist between response time and breadth of analysis. For example, in a recommendation system that generates real-time recommendations in response to requests from users, it may not be feasible to compare the user's ratings profile to those of all other users. A relatively shallow analysis of the available data (leading to poor recommendations) may therefore be performed.

Another problem with both collaborative and content-based systems is that they generally do not reflect the current preferences of the community of users. In the context of a system that recommends products to customers, for example, there is typically no mechanism for favoring items that are currently "hot sellers." In addition, existing systems do not provide a mechanism for recognizing that the user may be searching for a particular type or category of item.

SUMMARY OF THE DISCLOSURE

The present invention addresses these and other problems by providing a computer-implemented service and associated methods for generating personalized recommendations of items based on the collective interests of a community of users. An important benefit of the service is that the recommendations are generated without the need for the user, or any other users, to rate items. Another important benefit is that the recommended items are identified using a previously-generated table or other mapping structure which maps individual items to lists of "similar" items. The item similarities reflected by the table are based at least upon correlations between the interests of users in particular items.

The types of items that can be recommended by the service include, without limitation, books, compact discs ("CDs"), videos, authors, artists, item categories, Web sites, and chat groups. The service may be implemented, for example, as part of a Web site, online services network, e-mail notification service, document filtering system, or other type of computer system that explicitly or implicitly recommends items to users. In a preferred embodiment described herein, the service is used to recommend works such as book titles and music titles to users of an online merchant's Web site.

In accordance with one aspect of the invention, the mappings of items to similar items ("item-to-item mappings") are generated periodically, such as once per week, by an off-line process which identifies correlations between known interests of users in particular items. For example, in the embodiment described in detail below, the mappings are generating by periodically analyzing user purchase histories to identify correlations between purchases of items. The similarity between two items is preferably measured by determining the number of users that have an interest in both items relative to the number of users

that have an interest in either item (e.g., items A and B are highly similar because a relatively large portion of the users that bought one of the items also bought the other item). The item-to-item mappings could also incorporate other types of similarities, including content-based similarities extracted by analyzing item descriptions or content.

To generate a set of recommendations for a given user, the service retrieves from the table the similar items lists corresponding to items already known to be of interest to the user, and then appropriately combines these lists to generate a list of recommended items. For example, if there are three items that are known to be of interest to the user (such as three items the user recently purchased), the service may retrieve the similar items lists for these three items from the table and combine these lists. Because the item-to-item mappings are regenerated periodically based on up-to-date sales data, the recommendations tend to reflect the current buying trends of the community.

In accordance with another aspect of the invention, the similar items lists read from the table may be appropriately weighted (prior to being combined) based on indicia of the user's affinity for, or current interest in, the corresponding items of known interest. For example, the similar items list for a book that was purchased in the last week may be weighted more heavily than the similar items list for a book that was purchased four months ago. Weighting a similar items list heavily has the effect of increasing the likelihood that the items in that list will be included in the recommendations that are ultimately presented to the user.

An important aspect of the service is that the relatively computation-intensive task of correlating item interests is performed off-line, and the results of this task (item-to-item mappings) stored in a mapping structure for subsequent look-up. This enables the personal recommendations to be generated rapidly and efficiently (such as in real-time in response to a request by the user), without sacrificing breadth of analysis.

Another feature of the invention involves using the current and/or recent contents of the user's shopping cart as inputs to the recommendation service (or to another type of recommendation service which generates recommendations given a unary listing of items). For example, if the user currently has three items in his or her shopping cart, these three items can be treated as the items of known interest for purposes of generating recommendations, in which case the recommendations may be generated and displayed automatically when the user views the shopping cart contents. Using the current and/or recent shopping cart contents as inputs tends to produce recommendations that are highly correlated to the current short-term interests of the user—even if these short term interest differ significantly from the user's general preferences. For example, if the user is currently searching for books on a particular topic and has added several such books to the shopping cart, this method will more likely produce other books that involve the same or similar topics.

One aspect of the invention is thus a computer-implemented method of recommending items to a user. The method comprises identifying a plurality of items that are currently in the user's shopping cart; and using the plurality of items in the user's shopping cart to generate a list of additional items that are predicted to be of interest to the user, wherein an additional item is selected for inclusion in the list based in-part upon whether that additional item is similar to more than one of the plurality of items in the user's shopping cart. The list of additional items is displayed to the user when the user views contents of the shopping cart.

Another aspect of the invention is a method of recommending products to a user. The method comprises generating a data structure which maps individual products to sets of related products in which product relatedness is determined based at least in-part on an automated analysis of user purchase histories of products. The method further comprises identifying a plurality of products that are currently in a shopping cart of a user. For each of the plurality of products, the data structure is accessed to identify a corresponding set of related products, to thereby identify a plurality of sets of related products. The related products are selected from the sets to recommend to the user based in part on whether a related product falls within more than one of the sets, such that products that are related to more than one of the products in the user's shopping cart tend to be recommended to the user over products related to only a single product in the shopping cart.

Another feature of the invention involves allowing the user to create multiple shopping carts under a single account (such as shopping carts for different family members), and generating recommendations that are specific to a particular shopping cart. For example, the user can be prompted to select a particular shopping cart (or set of shopping carts), and the recommendations can then be generated based on the items that were purchased from or otherwise placed into the designated shopping cart(s). This feature of the invention allows users to obtain recommendations that correspond to the role or purpose (e.g., work versus pleasure) of a particular shopping cart.

Two specific implementations of the service are disclosed, both of which generate personal recommendations using the same type of table. In the first implementation, the recommendations are based on the items that have recently been rated or purchased by the user. In the second implementation, the recommendations are based on the current shopping cart contents of the user.

BRIEF DESCRIPTION OF THE DRAWINGS

These and other features of the invention will now be described with reference to the drawings summarized below. These drawings and the associated description are provided to illustrate a preferred embodiment of the invention, and not to limit the scope of the invention.

FIG. 1 illustrates a Web site which implements a recommendation service which operates in accordance with the invention, and illustrates the flow of information between components.

FIG. 2 illustrates a sequence of steps that are performed by the recommendation process of FIG. 1 to generate personalized recommendations.

FIG. 3 illustrates a sequence of steps that are performed by the table generation process of FIG. 1 to generate a similar items table, and illustrates temporary data structures generated during the process.

FIG. 4 is a Venn diagram illustrating a hypothetical purchase history profile of three items.

FIG. 5 illustrates one specific implementation of the sequence of steps of FIG. 2.

FIG. 6 illustrates the general form of a Web pages used to present the recommendations of the FIG. 5 process to the user.

FIG. 7 illustrates another specific implementation of the sequence of steps of FIG. 2.

DETAILED DESCRIPTION OF PREFERRED EMBODIMENTS

The various features and methods of the invention will now be described in the context of a recommendation

service, including two specific implementations thereof, that is used to recommend book titles, music titles, video titles, and other types of items to individual users of the Amazon.com Web site. As will be recognized to those skilled in the art, the disclosed methods can also be used to recommend other types of items, including non-physical items. By way of example and not limitation, the disclosed methods can also be used to recommend authors, artists, categories or groups of titles, Web sites, chat groups, movies, television shows, downloadable content, restaurants, and other users.

Throughout the description, reference will be made to various implementation-specific details of the recommendation service, the Amazon.com Web site, and other recommendation services of the Web site. These details are provided in order to fully illustrate preferred embodiments of the invention, and not to limit the scope of the invention. The scope of the invention is set forth in the appended claims.

I. Overview of Web Site and Recommendation Services

The Amazon.com Web site includes functionality for allowing users to search, browse, and make purchases from an online catalog of several million book titles, music titles, video titles, and other types of items. Using a shopping cart feature of the site, users can add and remove items to/from a personal shopping cart which is persistent over multiple sessions. (As used herein, a "shopping cart" is a data structure and associated code which keeps track of items that have been selected by a user for possible purchase.) For example, a user can modify the contents of the shopping cart over a period of time, such as one week, and then proceed to a check out area of the site to purchase the shopping cart contents.

The user can also create multiple shopping carts within a single account. For example, a user can set up separate shopping carts for work and home, or can set up separate shopping carts for each member of the user's family. A preferred shopping cart scheme for allowing users to set up and use multiple shopping carts is disclosed in U.S. application Ser. No. 09/104,942, filed Jun. 25, 1998, titled METHOD AND SYSTEM FOR ELECTRONIC COMMERCE USING MULTIPLE ROLES, the disclosure of which is hereby incorporated by reference.

The site also implements a variety of different recommendation services for recommending book titles, music titles, and/or video titles to users. One such service, known as BookMatcher™, allows users to interactively rate individual books on a scale of 1–5 to create personal item ratings profiles, and applies collaborative filtering techniques to these profiles to generate personal recommendations. The BookMatcher service is described in detail in U.S. application Ser. No. 09/040,171 filed Mar. 17, 1998, the disclosure of which is hereby incorporated by reference. The site may also include associated services that allow users to rate other types of items, such as CDs and videos. As described below, the ratings data collected by the BookMatcher service and similar services is optionally incorporated into the recommendation processes of the present invention.

Another type of service is a recommendation service which operates in accordance with the invention. The service ("Recommendation Service") is preferably used to recommend book titles, music titles and/or videos titles to users, but could also be used in the context of the same Web site to recommend other types of items, including authors, artists, and groups or categories of titles. Briefly, given a unary listing of items that are "known" to be of interest to a user (e.g., a list of items purchased, rated, and/or viewed by the user), the Recommendation Service generates a list of additional items ("recommendations") that are predicted to be of interest to the user. (As used herein, the term "interest" refers generally to a user's liking of or affinity for an item; the term "known" is used to distinguish items for which the user has implicitly or explicitly indicated some level of interest from items predicted by the Recommendation Service to be of interest.)

The recommendations are generated using a table which maps items to lists of "similar" items ("similar items lists"), without the need for users to rate any items (although ratings data may optionally be used). For example, if there are three items that are known to be of interest to a particular user (such as three items the user recently purchased), the service may retrieve the similar items lists for these three items from the table, and appropriately combine these lists (as described below) to generate the recommendations.

In accordance with one aspect of the invention, the mappings of items to similar items ("item-to-item mappings") are generated periodically, such as once per week, from data which reflects the collective interests of the community of users. More specifically, the item-to-item mappings are generated by an off-line process which identifies correlations between known interests of users in particular items. For example, in the embodiment described in detail below, the mappings are generating by analyzing user purchase histories to identify correlations between purchases of particular items (e.g., items A and B are similar because a relatively large portion of the users that purchased item A also bought item B). The item-to-item mappings could also reflect other types of similarities, including content-based similarities extracted by analyzing item descriptions or content.

An important aspect of the Recommendation Service is that the relatively computation-intensive task of correlating item interests is performed off-line, and the results of this task (item-to-item mappings) are stored in a mapping structure for subsequent look-up. This enables the personal recommendations to be generated rapidly and efficiently (such as in real-time in response to a request by the user), without sacrificing breadth of analysis.

In accordance with another aspect of the invention, the similar items lists read from the table are appropriately weighted (prior to being combined) based on indicia of the user's affinity for or current interest in the corresponding items of known interest. For example, in one embodiment described below, if the item of known interest was previously rated by the user (such as through use of the BookMatcher service), the rating is used to weight the corresponding similar items list. Similarly, the similar items list for a book that was purchased in the last week may be weighted more heavily than the similar items list for a book that was purchased four months ago.

Another feature of the invention involves using the current and/or recent contents of the user's shopping cart as inputs to the Recommendation Service. For example, if the user currently has three items in his or her shopping cart, these three items can be treated as the items of known interest for purposes of generating recommendations, in which case the recommendations may be generated and displayed automatically when the user views the shopping cart contents. If the user has multiple shopping carts, the recommendations are preferably generated based on the contents of the shopping cart implicitly or explicitly designated by the user, such as the shopping cart currently being viewed. This method of generating recommendations can also be used within other types of recommendation systems, including content-based systems and systems that do not use item-to-item mappings.

Using the current and/or recent shopping cart contents as inputs tends to produce recommendations that are highly correlated to the current short-term interests of the user—even if these short term interests are not reflected by the user's purchase history. For example, if the user is currently searching for a father's day gift and has selected several books for prospective purchase, this method will have a tendency to identify other books that are well suited for the gift recipient.

Another feature of the invention involves generating recommendations that are specific to a particular shopping cart. This allows a user who has created multiple shopping carts to conveniently obtain recommendations that are specific to the role or purpose to the particular cart. For example, a user who has created a personal shopping cart for buying books for her children can designate this shopping cart to obtain recommendations of children's books. In one embodiment of this feature, the recommendations are generated based solely upon the current contents of the shopping cart selected for display. In another embodiment, the user may designate one or more shopping carts to be used to generate the recommendations, and the service then uses the items that were purchased from these shopping carts as the items of known interest.

As will be recognized by those skilled in the art, the above-described techniques for using shopping cart contents to generate recommendations can also be incorporated into other types of recommendation systems, including pure content-based systems.

FIG. **1** illustrates the basic components of the Amazon.com Web site **30**, including the components used to implement the Recommendation Service. The arrows in FIG. **1** show the general flow of information that is used by the Recommendation Service. As illustrated by FIG. **1**, the Web site **30** includes a Web server application **32** ("Web server") which processes HTTP (Hypertext Transfer Protocol) requests received over the Internet from user computers **34**. The Web server **34** accesses a database **36** of HTML (Hypertext Markup Language) content which includes product information pages and other browsable information about the various products of the catalog. The "items" that are the subject of the Recommendation Service are the titles (regardless of media format such as hardcover or paperback) that are represented within this database **36**.

The Web site **30** also includes a "user profiles" database **38** which stores account-specific information about users of the site. Because a group of individuals can share an account, a given "user" from the perspective of the Web site may include multiple actual users. As illustrated by FIG. **1**, the data stored for each user may include one or more of the following types of information (among other things) that can be used to generate recommendations in accordance with the invention: (a) the user's purchase history, including dates of purchase, (b) the user's item ratings profile (if any), (c) the current contents of the user's personal shopping cart(s), and (d) a listing of items that were recently (e.g., within the last six months) removed from the shopping cart(s) without being purchased ("recent shopping cart contents"). If a given user has multiple shopping carts, the purchase history for that user may include information about the particular shopping cart used to make each purchase; preserving such information allows the Recommendation Service to be configured to generate recommendations that are specific to a particular shopping cart.

As depicted by FIG. **1**, the Web server **32** communicates with various external components **40** of the site. These external components **40** include, for example, a search engine and associated database (not shown) for enabling users to interactively search the catalog for particular items. Also included within the external components **40** are various order processing modules (not shown) for accepting and processing orders, and for updating the purchase histories of the users.

The external components **40** also include a shopping cart process (not shown) which adds and removes items from the users' personal shopping carts based on the actions of the respective users. (The term "process" is used herein to refer generally to one or more code modules that are executed by a computer system to perform a particular task or set of related tasks.) In one embodiment, the shopping cart process periodically "prunes" the personal shopping cart listings of items that are deemed to be dormant, such as items that have not been purchased or viewed by the particular user for a predetermined period of time (e.g. two weeks). The shopping cart process also preferably generates and maintains the user-specific listings of recent shopping cart contents.

The external components **40** also include recommendation service components **44** that are used to implement the site's various recommendation services. Recommendations generated by the recommendation services are returned to the Web server **32**, which incorporates the recommendations into personalized Web pages transmitted to users.

The recommendation service components **44** include a BookMatcher application **50** which implements the above-described BookMatcher service. Users of the BookMatcher service are provided the opportunity to rate individual book titles from a list of popular titles. The book titles are rated according to the following scale:

1=Bad!
2=Not for me
3=OK
4=Liked it
5=Loved it!

Users can also rate book titles during ordinary browsing of the site. As depicted in FIG. **1**, the BookMatcher application **50** records the ratings within the user's items rating profile. For example, if a user of the BookMatcher service gives the book Into Thin Air a score of "5," the BookMatcher application **50** would record the item (by ISBN or other identifier) and the score within the user's item ratings profile. The BookMatcher application **50** uses the users' item ratings profiles to generate personal recommendations, which can be requested by the user by selecting an appropriate hyperlink. As described in detail below, the item ratings profiles are also used by an "Instant Recommendations" implementation of the Recommendation Service.

The recommendation services components **44** also include a recommendation process **52**, a similar items table **60**, and an off-line table generation process **66**, which collectively implement the Recommendation Service. As depicted by the arrows in FIG. **1**, the recommendation process **52** generates personal recommendations based on information stored within the similar items table **60**, and based on the items that are known to be of interest ("items of known interest") to the particular user.

In the embodiments described in detail below, the items of known interest are identified based on information stored in the user's profile, such as by selecting all items purchased by the user or all items in the user's shopping cart. In other embodiments of the invention, other types of methods or sources of information could be used to identify the items of known interest. For example, in a service used to recommend Web sites, the items (Web sites) known to be of

interest to a user could be identified by parsing a Web server access log and/or by extracting URLs from the "favorite places" list of the user's Web browser. In a service used to recommend restaurants, the items (restaurants) of known interest could be identified by parsing the user's credit card records to identify restaurants that were visited more than once.

The various processes **50**, **52**, **66** of the recommendation services may run, for example, on one or more Unix or NT based workstations or physical servers (not shown) of the Web site **30**. The similar items table **60** is preferably stored as a B-tree data structure to permit efficient look-up, and may be replicated across multiple machines (together with the associated code of the recommendation process **52**) to accommodate heavy loads.

II. Similar Items Table (FIG. 1)

The general form and content of the similar items table **60** will now be described with reference to FIG. **1**. As this table can take on many alternative forms, the details of the table are intended to illustrate, and not limit, the scope of the invention.

As indicated above, the similar items table **60** maps items to lists of similar items based at least upon the collective interests of the community of users. The similar items table **60** is preferably generated periodically (e.g., once per week) by the off-line table generation process **66**. The table generation process **66** generates the table **60** from data that reflects the collective interests of the community of users. In the embodiment described in detail herein, the similar items table is generated exclusively from the purchase histories of the community of users (as depicted in FIG. **1**). In other embodiments, the table **60** may additionally or alternatively be generated from other indicia of user-item interests, including indica based on users viewing activities, shopping cart activities, and item rating profiles. For example, the table **60** could be built exclusively from the present and/or recent shopping cart contents of users. The similar items table **60** could also reflect non-collaborative type item similarities, including content-based similarities derived by comparing item contents or descriptions.

Each entry in the similar items table **60** is preferably in the form of a mapping of a popular item **62** to a corresponding list **64** of similar items ("similar items lists"). As used herein, a "popular" item is an item which satisfies some pre-specified popularity criteria. For example, in the embodiment described herein, an item is treated as popular of it has been purchased by more than **30** customers during the life of the Web site. Using this criteria produces a set of popular items (and thus a recommendation service) which grows over time. The similar items list **64** for a given popular item **62** may include other popular items.

In other embodiments involving sales of products, the table **60** may include entries for most or all of the products of the online merchant, rather than just the popular items. In the embodiment described herein, several different types of items (books, CDs, videos, etc.) are reflected within the same table **60**, although separate tables could alternatively be generated for each type of item.

Each similar items list **64** consists of the N (e.g., 20) items which, based on correlations between purchases of items, are deemed to be the most closely related to the respective popular item **62**. Each item in the similar items list **64** is stored together with a commonality index ("CI") value which indicates the relatedness of that item to the popular item **62**, based on sales of the respective items. A relatively high commonality index for a pair of items ITEM A and ITEM B indicates that a relatively large percentage of users who bought ITEM A also bought ITEM B (and vice versa). A relatively low commonality index for ITEM A and ITEM B indicates that a relatively small percentage of the users who bought ITEM A also bought ITEM B (and vice versa). As described below, the similar items lists are generated, for each popular item, by selecting the N other items that have the highest commonality index values. Using this method, ITEM A may be included in ITEM B's similar items list even though ITEM B in not present in ITEM A's similar items list.

In the embodiment depicted by FIG. **1**, the items are represented within the similar items table **60** using product IDs, such as ISBNs or other identifiers. Alternatively, the items could be represented within the table by title ID, where each title ID corresponds to a given "work" regardless of its media format. In either case, different items which correspond to the same work, such as the hardcover and paperback versions of a given book or the VCR cassette and DVD versions of a given video, are preferably treated as a unit for purposes of generating recommendations.

Although the recommendable items in the described system are in the form of book titles, music titles and videos titles, it will be appreciated that the underlying methods and data structures can be used to recommend a wide range of other types of items. For example, in the system depicted by FIG. **1**, the Recommendation Service could also be used to recommend authors, artists, and categorizations or groups of works.

III. General Process for Generating Recommendations (FIG. **2**)

The general sequence of steps that are performed by the recommendation process **52** to generate a set of personal recommendations will now be described with reference to FIG. **2**. This process, and the more specific implementations of the process depicted by FIGS. **5** and **7** (described below), are intended to illustrate, and not limit, the scope of the invention.

The FIG. **2** process is preferably invoked in real-time in response to an online action of the user. For example, in an Instant Recommendations implementation (FIGS. **5** and **6**) of the service, the recommendations are generated and displayed in real-time (based on the user's purchase history and/or item ratings profile) in response to selection by the user of a corresponding hyperlink, such as a hyperlink which reads "Instant Book Recommendations" or "Instant Music Recommendations." In a shopping cart based implementation (FIG. **7**), the recommendations are generated (based on the user's current and/or recent shopping cart contents) in real-time when the user initiates a display of a shopping cart, and are displayed on the same Web page as the shopping cart contents. The Instant Recommendations and shopping cart based embodiments are described separately below under corresponding headings.

Any of a variety of other methods can be used to initiate the recommendations generation process and to display the recommendations to the user. For example, the recommendations can automatically be generated periodically and sent to the user by e-mail, in which case the e-mail listing may contain hyperlinks to the product information pages of the recommended items. Further, the personal recommendations could be generated in advance of any request or action by the user, and cached by the Web site **30** until requested.

As illustrated by FIG. **2**, the first step (step **80**) of the recommendations-generation process involves identifying a set of items that are of known interest to the user. The "knowledge" of the user's interest can be based on explicit indications of interest (e.g., the user rated the item highly) or

implicit indications of interest (e.g., the user added the item to a shopping cart). Items that are not "popular items" within the similar items table **60** can optionally be ignored during this step.

In the embodiment depicted in FIG. **1**, the items of known interest are selected from one or more of the following groups: (a) items in the user's purchase history (optionally limited to those items purchased from a particular shopping cart); (b) items in the user's shopping cart (or a particular shopping cart designated by the user), (c) items rated by the user (optionally with a score that exceeds a certain threshold, such as two), and (d) items in the "recent shopping cart contents" list associated with a given user or shopping cart. In other embodiments, the items of known interest may additionally or alternatively be selected based on the viewing activities of the user. For example, the recommendations process **52** could select items that were viewed by the user for an extended period of time and/or viewed more than once. Further, the user could be prompted to select items of interest from a list of popular items.

For each item of known interest, the service retrieves the corresponding similar items list **64** from the similar items table **60** (step **82**), if such a list exists. If no entries exist in the table **60** for any of the items of known interest, the process **52** may be terminated; alternatively, the process could attempt to identify additional items of interest, such as by accessing other sources of interest information.

In step **84**, the similar items lists **64** are optionally weighted based on information about the user's affinity for the corresponding items of known interest. For example, a similar items list **64** may be weighted heavily if the user gave the corresponding popular item a rating of "5" on a scale or 1–5, or if the user purchased multiple copies of the item. Weighting a similar items list **64** heavily has the effect of increasing the likelihood that the items in that list we be included in the recommendations that are ultimately presented to the user. In one implementation described below, the user is presumed to have a greater affinity for recently purchased items over earlier purchased items.

The similar items lists **64** are preferably weighted by multiplying the commonality index values of the list by a weighting value. The commonality index values as weighted by any applicable weighting value are referred to herein as "scores." In other embodiments, the recommendations may be generated without weighting the similar items lists **64**.

If multiple similar items lists **64** are retrieved in step **82**, the lists are appropriately combined (step **86**), such as by merging the lists while summing the scores of like items. The resulting list is then sorted (step **88**) in order of highest-to-lowest score. In step **90**, the sorted list is filtered to remove unwanted items. The items removed during the filtering process may include, for example, items that have already been purchased or rated by the user, and items that fall outside any product group (such as music or books), product category (such as non-fiction), or content rating (such as PG or adult) designated by the user. The filtering step could alternatively be performed at a different stage of the process, such as during the retrieval of the similar items lists from the table **60**. The result of step **90** is a list ("recommendations list") of other items to be recommended to the user.

In step **92**, one or more additional items are optionally added to the recommendations list. In one embodiment, the items added in step **92** are selected from the set of items (if any) in the user's "recent shopping cart contents" list. As an important benefit of this step, the recommendations include one or more items that the user previously considered purchasing but did not purchase. The items added in step **92** may additionally or alternatively be selected using another recommendations method, such as a content-based method.

Finally, in step **94**, a list of the top M (e.g., 15) items of the recommendations list are returned to the Web server **32** (FIG. **1**). The Web server incorporates this list into one or more Web pages that are returned to the user, with each recommended item being presented as a hypertextual link to the item's product information page. The recommendations may alternatively be conveyed to the user by email, facsimile, or other transmission method. Further, the recommendations could be presented as advertisements for the recommended items.

IV. Generation of Similar Items Table (FIGS. **3** and **4**)

The table-generation process **66** is preferably executed periodically (e.g., once a week) to generate a similar items table **60** that reflects the most recent purchase history data. The recommendation process **52** uses the most recently generated version of the table **60** to generate recommendations.

FIG. **3** illustrates the sequence of steps that are performed by the table generation process **66** to build the similar items table **60**. The general form of temporary data structures that are generated during the process are shown at the right of the drawing. As will be appreciated by those skilled in the art, any of a variety of alternative methods could be used to generate the table **60**.

As depicted by FIG. **3**, the process initially retrieves the purchase histories for all customers (step **100**). Each purchase history is in the general form of the user ID of a customer together with a list of the product IDs (ISBNs, etc.) of the items (books, CDs, videos, etc.) purchased by that customer. In embodiments which support multiple shopping carts within a given account, each shopping cart could be treated as a separate customer for purposes of generating the table. For example, if a given user (or group of users that share an account) purchased items from two different shopping carts within the same account, these purchases could be treated as the purchases of separate users.

The product IDs may be converted to title IDs during this process, or when the table **60** is later used to generate recommendations, so that different versions of an item (e.g., hardcover and paperback) are represented as a single item. This may be accomplished, for example, by using a separate database which maps product IDs to title IDs. To generate a similar items table that strongly reflects the current tastes of the community, the purchase histories retrieved in step **100** can be limited to a specific time period, such as the last six months.

In steps **102** and **104**, the process generates two temporary tables **102A** and **104A**. The first table **102A** maps individual customers to the items they purchased. The second table **104A** maps items to the customers that purchased such items. To avoid the effects of "ballot stuffing," multiple copies of the same item purchased by a single customer are represented with a single table entry. For example, even if a single customer purchased **4000** copies of one book, the customer will be treated as having purchased only a single copy. In addition, items that were sold to an insignificant number (e.g., <15) of customers are preferably omitted or deleted from the tables **102A**, **104B**.

In step **106**, the process identifies the items that constitute "popular" items. This may be accomplished, for example, by selecting from the item-to-customers table **104A** those items that were purchased by more than a threshold number (e.g., 30) of customers. In the context of the Amazon.com Web site, to resulting set of popular items may contain hundreds of thousands or millions of items.

In step **108**, the process counts, for each (popular_item, other_item) pair, the number of customers that are in common. A pseudocode sequence for performing this step is listed in Table 1. The result of step **108** is a table that indicates, for each (popular_item, other_item) pair, the number of customers the two have in common. For example, in the hypothetical table **108**A of FIG. 3, POPULAR_A and ITEM_B have seventy customers in common, indicating that seventy customers bought both items.

TABLE 1

```
for each popular_item
    for each customer in customers of item
        for each other_item in items of customer
            increment common-customer-count(popular_item,
            other_item)
```

In step **110**, the process generates the commonality indexes for each (popular_item, other_item) pair in the table **108**A. As indicated above, the commonality index (CI) values are measures of the similarity between two items, with larger CI values indicating greater degrees of similarity. The commonality indexes are preferably generated such that, for a given popular_item, the respective commonality indexes of the corresponding other_items take into consideration both (a) the number of customers that are common to both items, and (b) the total number of customers of the other_item. A preferred method for generating the commonality index values is set forth in the equation below.

$$CI(\text{item\_A, item\_B}) = \frac{\text{customers of item\_A and item\_B}}{\sqrt{(\text{customers of item\_A}) \times (\text{customers of item\_B})}}$$

FIG. **4** illustrates this method in example form. In the FIG. **4** example, item_P (a popular item) has two "other items," item_X and item_Y. Item_P has been purchased by 300 customers, item_X by 300 customers, and item_Y by 30,000 customers. In addition, item_P and item_X have 20 customers in common, and item_P and item_Y have 25 customers in common. Applying the equation above to the values shown in FIG. **4** produces the following results:

CI(item_P, item_X)=20/sqrt(300×300))=0.0667

CI(item_P, item_Y)=25/sqrt(300×30,000))=0.0083

Thus, even though items P and Y have more customers in common than items P and X, items P and X are treated as being more similar than items P and Y. This result desirably reflects the fact that the percentage of item_X customers that bought item_P (6.7%) is much greater than the percentage of item_Y customers that bought item_P (0.08%).

Because this equation is symmetrical (i.e., CI(item_A, item_B)=CI(item_B, item_A)), it is not necessary to separately calculate the CI value for every location in the table **108**A. In other embodiments, an asymmetrical method may be used to generate the CI values. For example, the CI value for a (popular_item, other_item) pair could be generated as (customers of popular_item and other_item)/(customers of other_item).

Following step **110** of FIG. **3**, each popular item has a respective "other_items" list which includes all of the other_items from the table **108**A and their associated CI values. In step **112**, each other_items list is sorted from highest-to-lowest commonality index. Using the FIG. **4** values as an example, item_X would be positioned closer to the top of the item_B's list than item_Y, since 0.014907>0.001643.

In step **114**, the sorted other_items lists are filtered by deleting all list entries that have fewer than 3 customers in common. For example, in the other_items list for POPULAR_A in table **108**A, ITEM_A would be deleted since POPULAR_A and ITEM_A have only two customers in common. Deleting such entries tends to reduce statistically poor correlations between item sales.

In step **116**, the sorted other_items lists are truncated to length N to generate the similar items lists, and the similar items lists are stored in a B-tree table structure for efficient look-up

As indicated above, any of a variety of other methods for evaluating similarities between items could be incorporated into the table generation process **66**. For example, the table generation process could compare item contents and/or use previously-assigned product categorizations as additional indicators of item similarities. An important benefit of the FIG. **3** method, however, is that the items need not contain any content that is amenable to feature extraction techniques, and need not be pre-assigned to any categories. For example, the method can be used to generate a similar items table given nothing more than the product IDs of a set of products and user purchase histories with respect to these products.

Another important benefit of the Recommendation Service is that the bulk of the processing (the generation of the similar items table **60**) is performed by an off-line process. Once this table has been generated, personalized recommendations can be generated rapidly and efficiently, without sacrificing breadth of analysis.

V. Instant Recommendations Service (FIGS. **5** and **6**)

A specific implementation of the Recommendation Service, referred to herein as the Instant Recommendations service, will now be described with reference to FIGS. **5** and **6**.

As indicated above, the Instant Recommendations service is invoked by the user by selecting a corresponding hyperlink from a Web page. For example, the user may select an "Instant Book Recommendations" or similar hyperlink to obtain a listing of recommended book titles, or may select a "Instant Music Recommendations" or "Instant Video Recommendations" hyperlink to obtain a listing of recommended music or video titles. As described below, the user can also request that the recommendations be limited to a particular item category, such as "non-fiction," "jazz" or "comedies." The Instant Recommendations service generates the recommendations based exclusively on the purchase history and any item ratings profile of the particular user. The service becomes available to the user (i.e., the appropriate hyperlink is presented to the user) once the user has purchased and/or rated a threshold number (e.g. three) of popular items within the corresponding product group. If the user has established multiple shopping carts, the user may also be presented the option of designating a particular shopping cart to be used in generating the recommendations.

FIG. **5** illustrates the sequence of steps that are performed by the Instant Recommendations service to generate personal recommendations. Steps **180**–**194** in FIG. **5** correspond, respectively, to steps **80**–**94** in FIG. **2**. In step **180**, the process **52** identifies all popular items that have been purchased by the user (from a particular shopping cart, if designated) or rated by the user, within the last six months. In step **182**, the process retrieves the similar items lists **64** for these popular items from the similar items table **60**.

In step **184**, the process **52** weights each similar items list based on the duration since the associated popular item was purchased by the user (with recently-purchased items

weighted more heavily), or if the popular item was not purchased, the rating given to the popular item by the user. The formula used to generate the weight values to apply to each similar items list is listed in C in Table 2. In this formula, "is_Purchased" is a boolean variable which indicates whether the popular item was purchased, "rating" is the rating value (1–5), if any, assigned to the popular item by the user, "order_date" is the date/time (measured in seconds since 1970) the popular item was purchased, "now" is the current date/time (measured in seconds since 1970), and "6 months" is six months in seconds.

TABLE 2

```
1 Weight = ( (is_purchased ? 5 : rating) * 2 - 5) *
2 ( 1 + (max( (is purchased ? order_date : 0) - (now - 6 months), 0) )
3 / (6 months))
```

In line 1 of the formula, if the popular item was purchased, the value "5" (the maximum possible rating value) is selected; otherwise, the user's rating of the item is selected. The selected value (which may range from 1–5) is then multiplied by 2, and 5 is subtracted from the result. The value calculated in line 1 thus ranges from a minimum of −3 (if the item was rated a "1") to a maximum of 5 (if the item was purchased or was rated a "5").

The value calculated in line 1 is multiplied by the value calculated in lines 2 and 3, which can range from a minimum of 1 (if the item was either not purchased or was purchased at least six months ago) to a maximum of 2 (if order_date=now). Thus, the weight can range from a minimum of −6 to a maximum of 10. Weights of zero and below indicate that the user rated the item a "2" or below. Weights higher than 5 indicate that the user actually purchased the item (although a weight of 5 or less is possible even if the item was purchased), with higher values indicating more recent purchases.

The similar items lists **64** are weighted in step **184** by multiplying the CI values of the list by the corresponding weight value. For example, if the weight value for a given popular item is ten, and the similar items list **64** for the popular item is

(productid_A, 0.10), (productid_B, 0.09), (productid_C, 0.08), . . .

the weighted similar items list would be:

(productid_A, 1.0), (productid_B, 0.9), (productid_C, 0.8), . . .

The numerical values in the weighted similar items lists are referred to as "scores."

In step **186**, the weighted similar items lists are merged (if multiple lists exist) to form a single list. During this step, the scores of like items are summed. For example, if a given other_item appears in three different similar items lists **64**, the three scores (including any negative scores) are summed to produce a composite score.

In step **188**, the resulting list is sorted from highest-to-lowest score. The effect of the sorting operation is to place the most relevant items at the top of the list. In step **190**, the list is filtered by deleting any items that (1) have already been purchased or rated by the user, (2) have a negative score, or (3) do not fall within the designated product group (e.g., books) or category (e.g., "science fiction," or "jazz").

In step **192** one or more items are optionally selected from the recent shopping cart contents list (if such a list exists) for the user, excluding items that have been rated by the user or which fall outside the designated product group or category. The selected items, if any, are inserted at randomly-selected locations within the top M (e.g., 15) positions in the recommendations list. Finally, in step **194**, the top M items from the recommendations list are returned to the Web server **32**, which incorporates these recommendations into one or more Web pages.

The general form of such a Web page is shown in FIG. **6**, which lists five recommended items. From this page, the user can select a link associated with one of the recommended items to view the product information page for that item. In addition, the user can select a "more recommendations" button **200** to view additional items from the list of M items. Further, the user can select a "refine your recommendations" link to rate or indicate ownership of the recommended items. Indicating ownership of an item causes the item to be added to the user's purchase history listing.

The user can also select a specific category such as "non-fiction" or "romance" from a drop-down menu **202** to request category-specific recommendations. Designating a specific category causes items in all other categories to be filtered out in step **190** (FIG. **5**).

VI. Shopping Cart Based Recommendations (FIG. 7)

Another specific implementation of the Recommendation Service, referred to herein as shopping cart recommendations, will now be described with reference to FIG. 7.

The shopping cart recommendations service is preferably invoked automatically when the user displays the contents of a shopping cart that contains more than a threshold number (e.g., 1) of popular items. The service generates the recommendations based exclusively on the current contents of the shopping cart. As a result, the recommendations tend to be highly correlated to the user's current shopping interests. In other implementations, the recommendations may also be based on other items that are deemed to be of current interest to the user, such as items in the recent shopping cart contents of the user and/or items recently viewed by the user. Further, other indications of the user's current shopping interests could be incorporated into the process. For example, any search terms typed into the site's search engine during the user's browsing session could be captured and used to perform content-based filtering of the recommended items list.

FIG. **7** illustrates the sequence of steps that are performed by the shopping cart recommendations service to generate a set of shopping-cart-based recommendations. In step **282**, the similar items list for each popular item in the shopping cart is retrieved from the similar items table **60**. The similar items list for one or more additional items that are deemed to be of current interest could also be retrieved during this step, such as the list for an item recently deleted from the shopping cart or recently viewed for an extended period of time.

In step **286**, these similar items lists are merged while summing the commonality index (CI) values of like items. In step **288**, the resulting list is sorted from highest-to-lowest score. In step **290**, the list is filtered to remove any items that exist in the shopping cart or have been purchased or rated by the user. Finally, in step **294**, the top M (e.g., 5) items of the list are returned as recommendations. The recommendations are preferably presented to the user on the same Web page (not shown) as the shopping cart contents.

If the user has defined multiple shopping carts, the recommendations generated by the FIG. **7** process may be based solely on the contents of the shopping cart currently selected for display. As described above, this allows the user to obtain recommendations that correspond to the role or purpose of a particular shopping cart (e.g., work versus home).

The various uses of shopping cart contents to generate recommendations as described above can be applied to other types of recommendation systems, including content-based systems. For example, the current and/or past contents of a shopping cart can be used to generate recommendations in a system in which mappings of items to lists of similar items are generated from a computer-based comparison of item contents. Methods for performing content-based similarity analyses of items are well known in the art, and are therefore not described herein.

Although this invention has been described in terms of certain preferred embodiments, other embodiments that are apparent to those of ordinary skill in the art are also within the scope of this invention. For example, although the embodiments described herein employ item lists, other programming methods for keeping track of and combining sets of similar items can be used. Accordingly, the scope of the present invention is intended to be defined only by reference to the appended claims.

In the claims which follow, reference characters used to denote process steps are provided for convenience of description only, and not to imply a particular order for performing the steps.

What is claimed is:

1. In a computer system that provides user access to a database of items, and provides electronic shopping carts for allowing users to interactively select and hold items from the database for prospective purchase, a system for recommending items to users, comprising:

a computer-readable medium embodying a non-user-specific data structure which maps items from the database to sets of similar items from the database; and

a computer system embodying a recommendation process which generates personal recommendations for a user that has an electronic shopping cart by at least:

(a) identifying a plurality of items, including multiple items that are currently in the user's shopping cart;

(b) for each item identified in step (a), accessing the non-user-specific data structure to identify a corresponding set of similar items, to thereby identify a plurality of sets of similar items;

(c) combining the sets of similar items identified in step (b) to generate a ranked set of similar items in which a similar item's ranking reflects whether that similar item appears within more than one of said sets; and

(d) presenting at least some of the items of the ranked set of similar items to the user as recommendations.

2. The system of claim 1, wherein step (a) consists of identifying items that are currently in the user's shopping cart.

3. The system of claim 1, wherein the shopping cart used in step (a) is a designated one of a plurality of shopping carts defined by the user within an account of the user, the system thereby generating recommendations that correspond to a shopping role of the designated shopping cart.

4. The system of claim 3, wherein the shopping cart used in step (a) is a shopping cart selected by the user for display.

5. The system of claim 1, wherein the computer system allows a user to create multiple shopping carts within a single account, and the recommendation process generates shopping cart specific recommendations to allow a user with multiple shopping carts to obtain recommendations specific to a role of a particular shopping cart.

6. The system of claim 1, wherein the data structure maps items to similar items based at least upon correlations between purchases of items by users.

7. The system of claim 1, wherein the data structure maps items to similar items based at least upon a content-based similarity analysis of items.

8. The system of claim 1, wherein the sets of similar items in the data structure include similarity index values, each index value indicating a degree of similarity between a reference item and a similar item, and wherein step (c) comprises combining similarity index values of like items.

9. The system of claim 1, further comprising filtering out similar items identified in step (b) to remove items already purchased by the user.

10. The system of claim 1, wherein step (d) comprises displaying the recommendations to the user when the user displays the contents of the shopping cart.

11. In a computer system that provides user access to a database of items, and provides electronic shopping carts for allowing users to collect and hold items for prospective purchase, a method of predicting items that are of current interest to a user, comprising the computer-implemented steps of:

(a) identifying a plurality of items that are currently in the user's shopping cart;

(b) for each item identified in step (a), accessing a non-user-specific data structure which maps items to similar items to identify a corresponding set of similar items, to thereby identify multiple sets of similar items;

(c) combining the multiple sets of similar items to form a ranked set of similar items in which an item's ranking reflects whether that item appears within more than one of the multiple sets; and

(d) recommending at least some of the items of the ranked set of similar items to the user.

12. The method of claim 11, wherein step (d) comprises presenting recommended items to the user when the user views the shopping cart contents.

13. The method of claim 11, wherein step (b) comprises accessing a data structure that maps items to similar items based at least upon correlations between purchases of items by users.

14. The method of claim 11, wherein step (b) comprises accessing a data structure that maps items to similar items based at least upon a content-based similarity analysis of items.

15. The method of claim 11, wherein step (a) further comprises identifying at least one additional item for which the user has recently indicated an interest.

16. In a computer system that provides user access to a database of items, and provides electronic shopping carts for allowing users to collect and hold items for prospective purchase, a computer-implemented method of recommending items to a user, comprising:

(a) providing a non-user-specific data structure which indicates similarities between items of the database;

(b) identifying a first plurality of items that are of current interest to the user, the first plurality of items including at least two items that are currently in an electronic shopping cart of the user;

(c) using the data structure to identify a second plurality of items that are similar to one or more of the first plurality of items and

(d) selecting, from the second plurality of items, a subset of items to recommend to the user, wherein an item is selected based in part on whether it is similar to more than one of the first plurality of items.

17. The method of claim 16, wherein step (a) comprises generating the data structure, the step of generating the data structure comprising analyzing purchase history data of a community of users to identify correlations between purchases of items.

18. The method of claim 16, wherein step (a) comprises generating the data structure, the step of generating the data structure comprising performing a content-based analysis of items within the database.

19. The method of claim 16, further comprising presenting the subset of items to the user as recommendations when the user views contents of the shopping cart.

20. In a computer system that provides user access to a database of items that are available for purchase, and which allows users to create and use multiple shopping carts within a common account with a merchant to hold items for prospective purchase, a method of recommending items that correspond to a particular shopping cart of a user that has multiple shopping carts, comprising:

(a) providing a non-user-specific data structure which indicates similarities between items of the database;

(b) identifying a shopping cart of the plurality of shopping carts of the user;

(c) identifying a first plurality of items that have been placed by the user into the shopping cart identified in step (b);

(d) using the data structure to identify a second plurality of items that are similar to one or more of the first plurality of items; and

(e) selecting, from the second plurality of items, a subset of items to recommend to the user, wherein an item is selected based in part on whether that item is similar to more than one of the first plurality of items.

21. The method of claim 20, wherein step (b) comprises prompting the user to select a shopping cart from the plurality of shopping carts.

22. The method of claim 20, wherein step (b) comprises identifying a shopping cart currently selected by the user for display.

23. The method of claim 22, further comprising presenting the subset of items to the user as recommendations when the user views contents of the selected shopping cart.

24. The method of claim 20, wherein step (c) comprises identifying at least one item that was purchased by the user from the shopping cart identified in step (b).

25. The method of claim 20, wherein step (c) comprises identifying at least one item that is currently in the shopping cart identified in step (b).

26. The method of claim 20, wherein step (a) comprises generating the data structure, the step of generating the data structure comprising analyzing purchase history data of a community of users to identify correlations between purchases of items.

27. The method of claim 20, wherein step (a) comprises generating the data structure, the step of generating the data structure comprising performing a content-based analysis of items within the database.

28. A computer-implemented method of recommending items to a user, comprising:

identifying a plurality of items that are currently in the user's shopping cart;

using the plurality of items in the user's shopping cart to generate a list of additional items that are predicted to be of interest to the user, wherein an additional item is selected for inclusion in the list based in-part upon whether that additional item is similar to more than one of the plurality of items in the user's shopping cart; and

displaying the list of additional items to the user when the user views contents of the shopping cart.

29. The method as in claim 28, wherein using the plurality of items to generate a list comprises:

(a) for each of the plurality of items, accessing a data structure which maps items to sets of similar items to identify a corresponding set of similar items; and

(b) combining the sets of similar items identified in (a) to generate a ranked set of items in which items are ranked according to similarity to the plurality of items in the shopping cart.

30. The method as in claim 29, wherein (b) comprises increasing a ranking of an item that is similar to more than one of the plurality of items in the shopping cart.

31. The method as in claim 28, wherein using the plurality of items to generate a list comprises determining whether an item is similar to more than one of the plurality of items in the shopping cart.

32. The method as in claim 28, wherein using the plurality of items to generate a list comprises filtering out items that have been purchased by the user.

33. The method as in claim 28, wherein the shopping cart is one of multiple shopping carts within an account of the user, and the method is performed separately for each of the multiple shopping carts to provide shopping cart specific recommendations.

34. The system of claim 1, wherein the recommendation process performs (b)–(d) when the user views contents of the shopping cart.

35. The method of claim 11, wherein steps (b)–(d) are performed when the user views contents of the shopping cart.

36. The method of claim 16, wherein (b) and (c) are performed when the user views contents of the shopping cart, to thereby identify items to display to the user in conjunction with the contents of the shopping cart.

37. The method of claim 28, wherein the list of additional items is generated when the user selects the shopping cart for viewing.

38. The method of claim 29, wherein (a) and (b) are performed when the user views contents of the shopping cart.

39. The method of claim 16, wherein the first plurality of items consists of items currently in the user's shopping cart.

40. A method of recommending products to a user, comprising:

generating a data structure which maps individual products to sets of related products in which product relatedness is determined based at least in-part on an automated analysis of user purchase histories of products;

identifying a plurality of products that are currently in a shopping cart of a user;

for each of the plurality of products, accessing the data structure to identify a corresponding set of related products, to thereby identify a plurality of sets of related products; and

selecting related products from the plurality of sets to recommend to the user based in part on whether a related product falls within more than one of said sets, such that products that are related to more than one of the products in the user's shopping cart tend to be recommended to the user over products related to only a single product in the shopping cart.

41. The method of claim 40, wherein the data structure is generated in an off-line mode.

42. The method of claim 40, further comprising displaying the related products selected for recommendation within a web page that displays current contents of the shopping cart.

* * * * *

92